



Improving the Jacobian free Newton–Krylov method for the viscous–plastic sea ice momentum equation

Clint Seinen, Boualem Khouider*

The University of Victoria, Department of Mathematics and Statistics, Canada



HIGHLIGHTS

- Fast and accurate numerical schemes for the viscous–plastic sea ice equations, based on a Jacobian free Newton–Krylov (JFNK) algorithm.
- Using a fully second order Crank–Nicolson-type method to discretize the sea ice momentum equations.
- Second order approximation of the Jacobian matrix based on Gateaux differentiation.
- Newton iterations failed to achieve quadratic convergence perhaps due to inherent instabilities of the sea-ice equations is discussed.
- Presented strategy of conditional termination to achieve acceptable accuracy when Newton method fails to converge.

ARTICLE INFO

Article history:

Available online 4 October 2017

Keywords:

Sea ice dynamics
Viscous–plastic rheology
Crank–Nicolson
Jacobian-free
Newton–Krylov
Second order convergence

ABSTRACT

Sea ice plays a central role in regulating Earth's radiative budget because of its high albedo effect, and its melting level during the summer season is considered to be an important index of global warming. Contemporary earth system models (ESM) utilize complex dynamical models for the pack ice to account for the variations of sea ice cover and its feedback on the earth system. There is a wide consensus in the climate modeling community that the ice pack is most accurately modeled as a viscous–plastic flow with a highly nonlinear rheology consisting of an elliptic–yield–curve constitutive law. However, sea ice dynamics remains one of the most uncertain factors in the ESM's ability to address the climate change problem. The difficulty in accurately and efficiently solving numerically the associated highly nonlinear partial differential equations is believed to be a big contributor to this uncertainty. This work builds on recent efforts to construct fast and accurate numerical schemes for the viscous–plastic sea ice equations, based on a Jacobian free Newton–Krylov (JFNK) algorithm. Here, we propose to improve on the JFNK approach by using a fully second order Crank–Nicolson-type method to discretize the sea ice momentum equations (SIME) instead of the previously used first order backward Euler. More importantly, we improve on the Jacobian free approximation by expressing the derivatives of the least cumbersome and linear terms in the discretized SIME functional in closed form and use a second order Gateaux-derivative approximation for the remaining terms, instead of using a first order approximation of the Gateaux derivative for the whole Jacobian matrix. Numerical tests performed on a synthetic exact solution for an augmented set of equations demonstrated that the new scheme is indeed second order accurate and the second order approximation of the Jacobian matrix was revealed to be crucial for the convergence of the nonlinear solver. One of the main difficulties in the JFNK approach resides in deciding on a stopping criterion for the Newton iterations. Our tests show that iterating beyond a certain level can in fact deteriorate the solution and prevent convergence. To overcome this issue, we suggest to use a conditional termination strategy by stopping the iterations as soon as the residual starts to increase. The resulting gain in efficiency overshadows any gains in accuracy when requiring formal convergence.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

The rapid decline of Arctic summer sea ice cover observed during the last few decades is one of the most alarming impacts of climate change [1,2]. While most climate models agree on a

* Corresponding author.

E-mail addresses: cseinen@uvic.ca (C. Seinen), khouider@uvic.ca (B. Khouider).

summer-ice-free Arctic in the near future, they disagree on its timing as they disagree on the regional and inter-annual and seasonal variability of sea ice coverage and thickness [1,3]. Sea ice cover plays a key role in the climate system not only in terms of its high albedo effect that limits the amount of solar radiation absorbed by the ocean surface but also because it acts as a non trivial interface between the atmosphere and ocean in the polar regions [4].

The sea ice cover is formed by individual ice floes that move around and collide with each other under the highly irregular wind and ocean current stresses which lead to the formations of ridges and leads (regions of open water) due, respectively, to converging and diverging stresses [5]. This directly influences the distribution of the ice thickness and ice coverage which in turn affects the thermodynamics of the ice pack, especially the processes of melting and freezing as well as its solar insulation properties [6].

On scales much larger than the size of a typical ice floe, i.e. 25–50 km, the ice pack is viewed as a continuum flow of granular-type material undergoing deformations due to its own inertial forces in response to the external stresses [7]. Locally, the pack ice is regarded as a plastic material with a stress-independent strain rate [8], on the grounds that ice floes undergo irreversible deformations, leading to the formation of leads and ridges, when a critical stress threshold is reached. Based on this argument, Hibler [9] derived a constitutive law for sea ice rheology leading to the widely accepted viscous–plastic isotropic continuum modeling framework for the pack ice at large temporal and spatial scales, on the order of a few days and about a 100 km, respectively. Hibler's work set the theoretical basis for the general view of the pack ice as a two-dimensional thin layer of coherent ice floes separated by open water and air [4,8,10–13]. As done in many previous studies, in this work we adopt Hibler's viscous–plastic isotropic model for ice dynamics [6,7,14–17]; models with anisotropic corrections have emerged during the recent years to take into account the plasticity of ice floes at small scales [5,18] but exploring these developments has been left for future work.

Despite its popularity, Hibler's ice rheology is based on a constitutive law involving an elliptic yield curve leading to a highly nonlinear and highly stiff set of partial differential equations that are very challenging numerically [15,19]. The numerical stiffness arises because by design the elliptic yield curve allows the modeled ice to have very large resistance thresholds to compression and shear stresses but basically zero resistance to tensile stresses [19]. Hibler [6] and Zhang and Hibler [15] used a multi-time level method to solve the viscous–plastic (VP) equations, where the velocities were first advanced by implicitly solving the linearized system produced via a backward Euler step. The nonlinear coefficients were then updated and the process repeated a few times until a satisfactory convergence was reached; the solver was then advanced to the next time step. The associated pseudo-time iterations are referred to as the outer loop (OL) in opposition the iterations associated with solving the linearized system.

Later work by Hunke and Zhang [20] showed that despite the observed near instantaneous reaction, this implicit approach resulted in a very slow transient response to forcing unless small time steps, relative to the time-scale of the changing forcing, were used. More recently, it was shown by Lemieux and Tremblay [16] that these problems could be attributed to poor nonlinear convergence and that they could be remedied through many pseudo-time steps, or OL iterations. These problems in effect hindered the efficiency of the numerical scheme, making the method impractical. As another approach, to overcome the inherent numerical difficulties, Hunke and Dukowicz [17] (see also [21]) introduced a relaxation-type correction by making the stress term an extra-prognostic variable, allowing the use of a fully explicit scheme with fairly large time steps. The relaxation term however induces some elasticity into the system, allowing the ice to resist tensile

stresses and introduce short wavelength waves into the system. While the physicality of the elastic–viscous–plastic (EVP) ice model is debatable, the presence of the elastic waves induce a whole other numerical challenge [19].

This conundrum led Lemieux et al. [22] to propose a fully implicit scheme for the VP ice equations based on a Jacobian free Newton–Krylov algorithm, where the nonlinearity is tackled head-on using a pseudo–Newton scheme based on an approximate Jacobian matrix, built on a first order Gateaux derivative approximation, and compounded by a Krylov-type subspace approximation to solve the associated linear system at each pseudo–Newton iterate. Because the elliptic yield sometimes involves infinitely high viscosities that are typically capped by a large but fixed upper bound during a numerical simulation [6,15], Lemieux et al. [22] used a hyperbolic tangent smoother to make the rheology differentiable. The convergence properties of the Jacobian free Newton–Krylov (JFNK) scheme were put to the test by Lemieux et al. [19] who found that the new scheme is indeed superior to the EVP method, especially at high grid resolutions of 10–20 km because of the artificial amplification of grid scale elastic waves. However, the use of a first order backward Euler method and a first order Gateaux derivative approximation for the Jacobian matrix are less than optimal. Moreover, Lemieux et al. [16,19] used Taylor expansions to assign boundary conditions for the viscosities which could lead to incompatibilities with those of the prognostic variables, namely, the ice velocities and ice coverage and ice thickness.

Here we propose to improve on the JFNK method, which is indeed a fast and accurate alternative to existing methods, by using (1) a second order Crank–Nicolson scheme instead of the first order backward Euler scheme, (2) an improved approximation of the Jacobian matrix, namely by combining a differentiation in closed form for all the linear and nearly linear terms and a second order approximation of the Gateaux differential for the remaining nonlinear terms, and (3) writing the boundary conditions for the viscosities in terms of those of the prognostic variables. We then demonstrate the formal second order convergence of the improved JFNK scheme for a synthetic test solution, obtained by adding an artificial right hand side to the governing equations. We also analyze the sensitivity of the numerical solution to changes in a few key parameters of the method, namely those associated with the stopping criteria and especially the parameter ϵ involved in the approximation of the Gateaux differential. We found in particular, for the test case considered here, that while the first order approximation of the Gateaux derivative is very sensitive to the value of ϵ , the convergence of the present scheme, with the second order counterpart, is insensitive over the same range of ϵ values. Nonetheless, we also found that the method fails to reach convergence a couple of times, at coarse resolution, within a reasonable number of Newton iterations (about 10 or so). Our tests show that iterating beyond a certain level can in fact deteriorate the solution, causing larger errors and sometimes preventing convergence. To overcome this issue, we suggest to use a conditional termination strategy by stopping the iterations as soon as the residual starts to increase. The resulting gain in efficiency overshadows any gains in accuracy when requiring formal convergence.

The rest of the paper is organized as follows. In Section 2, we present the viscous–plastic sea ice momentum equations, with an elliptic yield curve using a hyperbolic tangent to smooth the viscosities. In Section 3, we discuss our basic numerical discretization strategy based on an Arakawa C-grid and introduce the three improvements discussed above. The main numerical test results are presented in Section 4, including the time evolution of the L_2 and L_∞ errors and the convergence under grid refinement as well as the sensitivity of the results to a few key model parameters.

2. The viscous–plastic sea ice momentum equation

The evolution of the large scale sea ice dynamics is primarily governed by the two-dimensional sea ice momentum equation (SIME) [8,23],

$$\rho \frac{D\mathbf{u}}{Dt} = -\rho h f (\mathbf{k} \times \mathbf{u}) + \tau_a - \tau_w + \nabla \cdot \boldsymbol{\sigma} - \rho h g \nabla H_d, \quad (2.1)$$

where, $\mathbf{u} = u\mathbf{i} + v\mathbf{j}$ is the horizontal sea ice velocity and \mathbf{i}, \mathbf{j} , and \mathbf{k} are the Cartesian unit vectors. We denote by x and y the coordinates in the east–west or zonal direction \mathbf{i} and in the north–south or meridional direction \mathbf{j} , respectively, while t is time. ρ and h represent the sea ice density and thickness, respectively, while f is the Coriolis parameter. τ_a and τ_w are the wind stress and water drag forcing terms. $\boldsymbol{\sigma}$ is the internal sea ice stress tensor and $\nabla \cdot \boldsymbol{\sigma}$ is known as the rheology term. H_d is the sea surface height and g is the acceleration due to gravity. In (2.1), $\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + u \frac{\partial}{\partial x} + v \frac{\partial}{\partial y}$ is the material derivative with respect to the ice motion. Because the pack ice velocities are relatively small, the contributions of the nonlinear advection terms in the SIME are typically neglected. We note that for the remainder of this paper, unless stated otherwise, bolded variables represent vector quantities.

As done in [23], the sea surface tilt, ∇H_d , is set by the geostrophic balance

$$-f\mathbf{k} \times \mathbf{u}_w^g = g \nabla H_d, \quad (2.2)$$

where \mathbf{u}_w^g is the geostrophic ocean current. Also following [19], the air and water drag terms are expressed via an empirical quadratic law with constant turning angles and constant drag coefficients,

$$\tau_a = \rho_a C_{da} |\mathbf{u}_a^g| (\mathbf{u}_a^g \cos \theta_a + (\mathbf{k} \times \mathbf{u}_a^g) \sin \theta_a), \quad (2.3)$$

$$\tau_w = C_w ((\mathbf{u} - \mathbf{u}_w^g) \cos \theta_w + (\mathbf{k} \times (\mathbf{u} - \mathbf{u}_w^g)) \sin \theta_w), \quad (2.4)$$

$$C_w = \rho_w C_{dw} |\mathbf{u} - \mathbf{u}_w^g| \quad (2.5)$$

where \mathbf{u}_a^g is the geostrophic wind and ρ_a and ρ_w represent the air and water densities, respectively, while C_{da} and C_{dw} are the air and water drag coefficients. Note that the sea ice velocity is neglected in the formulation of τ_a as $|\mathbf{u}_a^g| \gg |\mathbf{u}|$ [22].

Now, although the expression in (2.4)–(2.5) is nonlinear in \mathbf{u} , the key nonlinearity of the SIME comes from the formulation of the rheology term, $\nabla \cdot \boldsymbol{\sigma}$. We follow Hibler [6], and write the constitutive law, relating internal stresses and strain rates, as

$$\sigma_{ij} = 2\eta \dot{\epsilon}_{ij} + [\zeta - \eta] \dot{\epsilon}_{kk} \delta_{ij} - P \delta_{ij}/2, \quad i, j = 1, 2, \quad (2.6)$$

where σ_{ij} is the i, j component of the internal stress tensor, ζ and η are the bulk and shear viscosities, respectively, and δ_{ij} is the Kronecker delta. The strain rates, $\dot{\epsilon}_{ij}$, are defined as

$$\begin{aligned} \dot{\epsilon}_{11} &= \frac{\partial u}{\partial x}, \quad \dot{\epsilon}_{22} = \frac{\partial v}{\partial y}, \quad \dot{\epsilon}_{12} = \dot{\epsilon}_{21} = \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right), \\ \dot{\epsilon}_{kk} &= \dot{\epsilon}_{11} + \dot{\epsilon}_{22}. \end{aligned} \quad (2.7)$$

The parameterization of the ice strength or pressure, P , is formulated according to [6]

$$P = P^* h \exp [-C(1 - A)], \quad (2.8)$$

where $P^* = 27.5 \times 10^3 \text{ N m}^{-2}$ is the ice strength parameter and $C = 20$, the ice concentration parameter, is an empirical constant that characterizes the dependence of the compressive strength of ice on the thick ice area fraction or coverage A [22]; in [6], Hibler partitioned the pack ice into two distinctive areas of thick and thin ice, respectively.

In Hibler's original formulation [6], the viscosities are formulated using an elliptical yield curve with a normal flow rule, i.e.

$$\zeta = \frac{P}{2\Delta}, \quad \eta = \zeta e^{-2}, \quad (2.9)$$

where $e = 2$ is the principal axis ratio of the elliptical yield curve [6] and

$$\Delta = [(\dot{\epsilon}_{11}^2 + \dot{\epsilon}_{22}^2)(1 + e^{-2}) + 4e^{-2}\dot{\epsilon}_{12}^2 + 2\dot{\epsilon}_{11}\dot{\epsilon}_{22}(1 - e^{-2})]^{1/2}. \quad (2.10)$$

It is the combination of (2.6), (2.9), and (2.10) that leads to the very nonlinear nature of the SIME.

It is easy to see that when the strain rates go to zero, ζ and η can become arbitrarily large. To remedy this, during numerical simulations, Hibler [6] proposed the capping of the viscosities via

$$\zeta = \min \left(\frac{P}{2\Delta}, \zeta_{max} \right), \quad (2.11)$$

where $\zeta_{max} = kP$ and $k = 2.5 \times 10^8 \text{ s}$. Unfortunately, this method leads to a rheology term that is not continuously differentiable. A smooth formulation is beneficial for numerical purposes and Lemieux and Tremblay [16] have demonstrated faster convergence can be achieved when one is used. Therefore, we follow [16] and redefine the bulk viscosity as

$$\zeta \equiv \zeta_{max} \tanh \left(\frac{P}{2\Delta \zeta_{max}} \right) = kP \tanh \left(\frac{1}{2\Delta k} \right). \quad (2.12)$$

In addition to the momentum equations in (2.1), sea ice dynamics models also involve the continuity equations

$$\frac{\partial h}{\partial t} + \nabla \cdot (\mathbf{h}\mathbf{u}) = S_h, \quad (2.13)$$

and

$$\frac{\partial A}{\partial t} + \nabla \cdot (\mathbf{A}\mathbf{u}) = S_A, \quad (2.14)$$

which describe how the sea ice thickness, h , and area fraction of thick ice, A , are advected by the sea ice velocity field \mathbf{u} and altered according to the thermodynamic source/sink terms, S_h and S_A , due to melting and freezing, and the convergence and divergence of mass, which lead to the formation of leads and ridges. For this study, we focus on solving (2.1) only and thus the mass field is kept constant; the coupling of the momentum equations in (2.1) with (2.13) and (2.14) using the improved JFNK method proposed here is considered in [24].

3. Numerical approach

3.1. The rheology formulation

Prior to discussing the detailed numerical approach, we first expand the rheology term and write the u and v components of (2.1) as,

$$\begin{aligned} -\rho h \frac{\partial u}{\partial t} + \rho h f v + \tau_{au} - \tau_{wu} + \frac{\partial \sigma_{11}}{\partial x} + \frac{\partial \sigma_{12}}{\partial y} \\ - \rho h g \frac{\partial H_d}{\partial x} = 0, \end{aligned} \quad (3.1)$$

and

$$\begin{aligned} -\rho h \frac{\partial v}{\partial t} - \rho h f u + \tau_{av} - \tau_{wv} + \frac{\partial \sigma_{21}}{\partial x} + \frac{\partial \sigma_{22}}{\partial y} \\ - \rho h g \frac{\partial H_d}{\partial y} = 0, \end{aligned} \quad (3.2)$$

where τ_{au} and τ_{wu} represent the u -components of the air and water drag, and likewise for the v -components. Consistent with existing sea ice models [15,19,22,25], the advection nonlinearities are neglected in (3.1) and (3.2) because the ice velocities are typically small, as already mentioned.

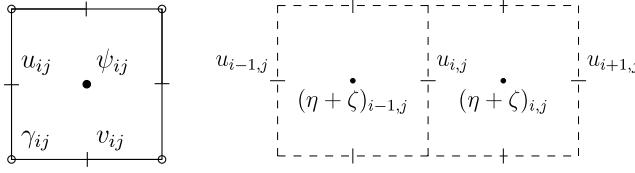


Fig. 3.1. The Arakawa C-grid (left) and an illustration of the stencil involved in the discretization of $\partial(\zeta + \eta)/\partial x$ at the u-grid point $u_{i,j}$ in (3.5) (right).

We note that the equations in (2.1) are written in non-conservative form. However, since h and ρ are kept constant and the advective nonlinearities are neglected, the method discussed here is independent on this particular formulation and its adaptation for the conservative formulation is straightforward, when h is allowed to vary.

The rheology terms in (3.1) and (3.2) are given by

$$\begin{aligned} \frac{\partial \sigma_{11}}{\partial x} + \frac{\partial \sigma_{12}}{\partial y} &= \frac{\partial}{\partial x} \left[(\eta + \zeta) \frac{\partial u}{\partial x} \right] + \frac{\partial}{\partial x} \left[(\zeta - \eta) \frac{\partial v}{\partial y} \right] \\ &\quad - \frac{1}{2} \frac{\partial P}{\partial x} + \frac{\partial}{\partial y} \left[\eta \frac{\partial u}{\partial y} \right] + \frac{\partial}{\partial y} \left[\eta \frac{\partial v}{\partial x} \right], \end{aligned} \quad (3.3)$$

and

$$\begin{aligned} \frac{\partial \sigma_{21}}{\partial x} + \frac{\partial \sigma_{22}}{\partial y} &= \frac{\partial}{\partial x} \left[\eta \frac{\partial u}{\partial y} \right] + \frac{\partial}{\partial x} \left[\eta \frac{\partial v}{\partial x} \right] + \frac{\partial}{\partial y} \left[(\eta + \zeta) \frac{\partial v}{\partial y} \right] \\ &\quad + \frac{\partial}{\partial y} \left[(\zeta - \eta) \frac{\partial u}{\partial x} \right] - \frac{1}{2} \frac{\partial P}{\partial y}. \end{aligned} \quad (3.4)$$

Following [22], we discretize the SIME using an Arakawa C-grid, where the ice velocities, u and v , are anchored at the centers of the vertical (along the y -coordinate) and the horizontal (along the x coordinate) edges, respectively, as illustrated in Fig. 3.1. For simplicity in exposition, the points at which the velocity u is defined are collectively called the u-grid while those associated with the v velocity are referred to as the v-grid.

On the C-grid, the cell centers are known as the tracer points, while the corner points are node points. Typically, scalar values like the ice strength, P , and the viscosities, η and ζ , are contained at the tracer points and interpolated to node points or the velocity grids as needed. This configuration allows for easy implementation of centered differences at the velocity grids. Using this approach, derivatives like $\frac{\partial}{\partial x}[(\eta + \zeta)\frac{\partial u}{\partial x}]$ in (3.3)–(3.4) are discretized according to the stencil in Fig. 3.1, amounting to

$$\begin{aligned} \frac{\partial}{\partial x} \left[(\eta + \zeta) \frac{\partial u}{\partial x} \right] \\ \approx \frac{(\eta + \zeta)_{i,j}[u_{i+1,j} - u_{i,j}] - (\eta + \zeta)_{i-1,j}[u_{i,j} - u_{i-1,j}]}{dx^2} \end{aligned} \quad (3.5)$$

at point (i, j) on the u-grid, denoted abusively as $u_{i,j}$, where dx represents the spatial resolution, which is assumed to be the same in both x - and y -directions. In this example, if $u_{i,j}$ is located at an ice/water boundary, the numerical derivative in (3.5) requires boundary conditions for the ice viscosities ζ and η , which are diagnostic variables completely determined by the ice velocities u , v and the ice strength P . However, due to the convoluted expression of Δ , deriving boundary conditions that are compatible with those eventually imposed on \mathbf{u} is not a trivial task; Lemieux et al. [22], for instance, use first order Taylor approximation to overcome this difficulty. To avoid this entirely, we propose expanding (3.3) and

(3.4) before their discretization, i.e.

$$\frac{\partial}{\partial x} \left[(\eta + \zeta) \frac{\partial u}{\partial x} \right] = (\partial_x \eta + \partial_x \zeta) \frac{\partial u}{\partial x} + (\eta + \zeta) \frac{\partial^2 u}{\partial x^2} \quad (3.6)$$

$$\frac{\partial}{\partial x} \left[(\zeta - \eta) \frac{\partial v}{\partial y} \right] = (\partial_x \zeta - \partial_x \eta) \frac{\partial v}{\partial y} + (\zeta - \eta) \frac{\partial^2 v}{\partial x \partial y} \quad (3.7)$$

$$\frac{\partial}{\partial y} \left[\eta \frac{\partial u}{\partial y} \right] = \partial_y \eta \frac{\partial u}{\partial y} + \eta \frac{\partial^2 u}{\partial y^2} \quad (3.8)$$

$$\frac{\partial}{\partial y} \left[\eta \frac{\partial v}{\partial x} \right] = \partial_y \eta \frac{\partial v}{\partial x} + \eta \frac{\partial^2 v}{\partial x \partial y} \quad (3.9)$$

$$\frac{\partial}{\partial x} \left[\eta \frac{\partial u}{\partial y} \right] = \partial_x \eta \frac{\partial u}{\partial y} + \eta \frac{\partial^2 u}{\partial x \partial y} \quad (3.10)$$

$$\frac{\partial}{\partial x} \left[\eta \frac{\partial v}{\partial x} \right] = \partial_x \eta \frac{\partial v}{\partial x} + \eta \frac{\partial^2 v}{\partial x^2} \quad (3.11)$$

$$\frac{\partial}{\partial y} \left[(\eta + \zeta) \frac{\partial v}{\partial y} \right] = (\partial_y \eta + \partial_y \zeta) \frac{\partial v}{\partial y} + (\eta + \zeta) \frac{\partial^2 v}{\partial y^2} \quad (3.12)$$

$$\frac{\partial}{\partial y} \left[(\zeta - \eta) \frac{\partial u}{\partial x} \right] = (\partial_y \zeta - \partial_y \eta) \frac{\partial u}{\partial x} + (\zeta - \eta) \frac{\partial^2 u}{\partial x \partial y}. \quad (3.13)$$

Moreover, to get $\partial_* \zeta$ and $\partial_* \eta$, where ∂_* represents a generic spatial derivative, we differentiate (2.12) to produce

$$\begin{aligned} \partial_* \zeta &= k \tanh \left(\frac{1}{2\Delta k} \right) \partial_* P - \frac{P}{2\Delta^2} \\ &\quad \times \left[1 - \tanh^2 \left(\frac{1}{2\Delta k} \right) \right] \partial_* \Delta, \end{aligned} \quad (3.14)$$

and for mathematical convenience, we use $\partial_* \Delta = \frac{\partial_*(\Delta^2)}{2\Delta}$ to get our final form

$$\begin{aligned} \partial_* \zeta &= k \tanh \left(\frac{1}{2\Delta k} \right) \partial_* P - \frac{P}{4\Delta^3} \\ &\quad \times \left[1 - \tanh^2 \left(\frac{1}{2\Delta k} \right) \right] \partial_* (\Delta^2), \end{aligned} \quad (3.15)$$

where we calculate $\partial_*(\Delta^2)$ according to

$$\begin{aligned} \partial_*(\Delta^2) &= (1 + e^{-2}) [2\dot{\epsilon}_{11}\partial_*\dot{\epsilon}_{11} + 2\dot{\epsilon}_{22}\partial_*\dot{\epsilon}_{22}] + 8e^{-2}\epsilon_{12}\partial_*\epsilon_{12} \\ &\quad + 2(1 - e^{-2}) [\dot{\epsilon}_{22}\partial_*\dot{\epsilon}_{11} + \dot{\epsilon}_{11}\partial_*\dot{\epsilon}_{22}]. \end{aligned} \quad (3.16)$$

Now, using (3.15) and (3.16), we no longer require ad-hoc boundary conditions for the viscosities, such as Taylor expansions. Finite differences are still utilized to calculate the spatial derivatives of P but this does not cause a problem since it has a rather simple expression in terms of h and A (2.8). It is worth noting that when $\Delta \rightarrow 0$, (3.15) becomes singular. To overcome this problem and avoid division by zero, when implementing this formula numerically, the machine epsilon is added to the denominator inside the hyperbolic tangent prior to differentiating (2.12).

3.2. A fully second order JFNK solver

3.2.1. The discretization

For our discretization, letting dt be our temporal resolution, we solve (3.1) and (3.2) at times dt , $2dt$, $3dt$, ..., ndt , ..., T , identifying variables at time level n via superscripts. Note that since the mass field is kept constant, both h and A are fixed in time. While previous implementations of this solver [19,22] use a backward Euler approach and achieve first order accuracy in time, we utilize the Crank–Nicolson discretization and center (3.1) and (3.2) at time level $n + 1/2$. The Crank–Nicolson discretization achieves second order accuracy in time via centered differences on

the temporal derivatives and the trapezoidal rule on the remaining terms, resulting in

$$\begin{aligned} & -\frac{\rho h}{dt} u^{n+1} + \frac{\rho hf}{2} v^{n+1} + \frac{1}{2} \tau_{au}^{n+1} - \frac{1}{2} \tau_{wu}^{n+1} + \frac{1}{2} \frac{\partial \sigma_{11}}{\partial x}^{n+1} \\ & + \frac{1}{2} \frac{\partial \sigma_{12}}{\partial y}^{n+1} - \frac{\rho hg}{2} \frac{\partial H_d}{\partial x}^{n+1} \\ & = -\frac{\rho h}{dt} u^n - \frac{\rho hf}{2} v^n - \frac{1}{2} \tau_{au}^n + \frac{1}{2} \tau_{wu}^n - \frac{1}{2} \frac{\partial \sigma_{11}}{\partial x}^n \\ & - \frac{1}{2} \frac{\partial \sigma_{12}}{\partial y}^n + \frac{\rho hg}{2} \frac{\partial H_d}{\partial x}^n \end{aligned} \quad (3.17)$$

and

$$\begin{aligned} & -\frac{\rho h}{dt} v^{n+1} - \frac{\rho hf}{2} u^{n+1} + \frac{1}{2} \tau_{av}^{n+1} - \frac{1}{2} \tau_{wv}^{n+1} + \frac{1}{2} \frac{\partial \sigma_{21}}{\partial x}^{n+1} \\ & + \frac{1}{2} \frac{\partial \sigma_{22}}{\partial y}^{n+1} - \frac{\rho hg}{2} \frac{\partial H_d}{\partial y}^{n+1} \\ & = -\frac{\rho h}{dt} v^n + \frac{\rho hf}{2} u^n - \frac{1}{2} \tau_{av}^n + \frac{1}{2} \tau_{wv}^n - \frac{1}{2} \frac{\partial \sigma_{21}}{\partial x}^n \\ & - \frac{1}{2} \frac{\partial \sigma_{22}}{\partial y}^n + \frac{\rho hg}{2} \frac{\partial H_d}{\partial y}^n. \end{aligned} \quad (3.18)$$

We then expand the rheology and water drag terms and group the right hand sides of the above equations into \mathcal{CN}_u and \mathcal{CN}_v and group the terms on the left hand sides independent of \mathbf{u} into r_u and r_v to get

$$\begin{aligned} & -\frac{\rho h}{dt} u^{n+1} + \frac{\rho hf}{2} v^{n+1} - \frac{1}{2} C_w^{n+1} (u^{n+1} \cos \theta_w - v^{n+1} \sin \theta_w) \\ & + \frac{1}{2} \frac{\partial}{\partial x} \left[(\eta + \zeta) \frac{\partial u}{\partial x} \right]^{n+1} + \frac{1}{2} \frac{\partial}{\partial x} \left[(\zeta - \eta) \frac{\partial v}{\partial y} \right]^{n+1} \\ & + \frac{1}{2} \frac{\partial}{\partial y} \left[\eta \frac{\partial u}{\partial y} \right]^{n+1} + \frac{1}{2} \frac{\partial}{\partial y} \left[\eta \frac{\partial v}{\partial x} \right]^{n+1} = \mathcal{CN}_u + r_u \end{aligned} \quad (3.19)$$

and

$$\begin{aligned} & -\frac{\rho h}{dt} v^{n+1} - \frac{\rho hf}{2} u^{n+1} - \frac{1}{2} C_w^{n+1} (v^{n+1} \cos \theta_w + u^{n+1} \sin \theta_w) \\ & + \frac{1}{2} \frac{\partial}{\partial x} \left[\eta \frac{\partial u}{\partial y} \right]^{n+1} \\ & + \frac{1}{2} \frac{\partial}{\partial x} \left[\eta \frac{\partial v}{\partial x} \right]^{n+1} + \frac{1}{2} \frac{\partial}{\partial y} \left[(\eta + \zeta) \frac{\partial v}{\partial y} \right]^{n+1} \\ & + \frac{1}{2} \frac{\partial}{\partial y} \left[(\zeta - \eta) \frac{\partial u}{\partial x} \right]^{n+1} = \mathcal{CN}_v + r_v. \end{aligned} \quad (3.20)$$

Keeping in mind our treatment of the rheology terms described in Section 3.1, it is at this point where we discretize the above equations spatially. For boundary conditions, we enforce Dirichlet conditions at land boundaries, $\mathbf{u} = \mathbf{0}$, and Neumann conditions at open-water boundaries, $\frac{\partial \mathbf{u}}{\partial \mathbf{n}} = \mathbf{0}$ (\mathbf{n} is the outward normal to the water–ice lateral boundary). As mentioned, we follow [19] and discretize (3.19)–(3.20) on the Arakawa C-grid (Fig. 3.1), achieving second order accuracy through centered differences. For our specific discretization, only the area fraction (A), ice thickness (h), and strength (P) are held at the tracer points; due to our treatment of the rheology term, the viscosities and their necessary derivatives are calculated directly at u and v points. Note that on the C-grid, u and v points are not collocated, and thus to populate v in the u equation of the SIME, and u for the v equation, we use a second order bi-linear interpolation based on the four surrounding points.

3.2.2. The solver

Implementing the discretization described above leads to a system of nonlinear equations of the form

$$A(\mathbf{u}^{n+1})\mathbf{u}^{n+1} = \mathbf{b}(\mathbf{u}^{n+1}), \quad (3.21)$$

where \mathbf{u}^{n+1} is the N dimensional solution vector, created by first stacking all u points in our computational domain and then all the v points, i.e.

$$\mathbf{u}^{n+1} = \begin{bmatrix} \mathbf{u}_u^{n+1} \\ \mathbf{u}_v^{n+1} \end{bmatrix},$$

where \mathbf{u}_u and \mathbf{u}_v are two vectors containing the grid point values of the u and v velocities, respectively, at the associated Arakawa C-grid points that are within the ice field, excluding grid-points that are entirely in open water to avoid unnecessary computations. Consequently, $A(\mathbf{u}^{n+1})$ is an $N \times N$ matrix containing all the necessary coefficients (both linear and nonlinear) from the left hand side of (3.19) and (3.20), and $\mathbf{b}(\mathbf{u}^{n+1})$ is an N dimensional vector containing \mathcal{CN}_u , \mathcal{CN}_v , r_u and r_v . We note that \mathbf{b} is a function of \mathbf{u}^{n+1} due to the presence of the nonlinear coefficient C_w^{n+1} . Moreover, since h and A are held constant, the ice is not moving in this study and thus N is fixed. However, when the ice moves and is allowed to melt and freeze, a moving mesh technique of certain sort needs to be adopted. One idea, currently under consideration by the authors and tested in [24], is to use a level-set method (commonly used in the simulation of multi-phase flows [26]) to track the ice–water interface with a signed distance function.

To solve (3.21), things are more involved than simply inverting the matrix, as both A and \mathbf{b} are nonlinear functions of \mathbf{u}^{n+1} . We must incorporate a nonlinear solver. For this we follow Lemieux et al. [19,22] and utilize a Jacobian free Newton–Krylov (JFNK) approach. The family of JFNK methods amount to applying Newton's method to a multivariate system, where the Newton updates are obtained by solving the resulting linear system of equations via a Krylov-type subspace approximation method. A good survey of the JFNK methods is found in [27], but we provide a brief description here for the sake of completeness.

Keeping in mind that we are solving for the solution at time level $n + 1$, we replace \mathbf{u}^{n+1} with a nonlinear iterate, \mathbf{u}^k , where the superscript now identifies the iteration number, and write the associated residual as

$$\mathbf{F}(\mathbf{u}^k) = A(\mathbf{u}^k)\mathbf{u}^k - \mathbf{b}(\mathbf{u}^k). \quad (3.22)$$

Defining $\delta \mathbf{u}^k = \mathbf{u}^{k+1} - \mathbf{u}^k$, we take a first order, multivariate Taylor Expansion about $\mathbf{F}(\mathbf{u}^k)$,

$$\mathbf{F}(\mathbf{u}^{k+1}) \approx \mathbf{F}(\mathbf{u}^k) + J(\mathbf{u}^k) \delta \mathbf{u}^k, \quad (3.23)$$

where $J(\mathbf{u}^k)$ is the Jacobian Matrix of $\mathbf{F}(\mathbf{u}^k)$, with respect to \mathbf{u}^k . Analogous to single variable Newton methods, the left hand side is set to zero to obtain the following linear system

$$J(\mathbf{u}^k) \delta \mathbf{u}^k = -\mathbf{F}(\mathbf{u}^k), \quad (3.24)$$

which is solved via a Krylov Subspace method to produce the update vector. For our JFNK method we use Intel's MKL GMRES routine to solve the system in (3.24), which is actually a Flexible GMRES (FGMRES) algorithm. As discussed in [28], this variant of the GMRES algorithm allows for right preconditioning with a dynamic preconditioner but despite this capability, this is not utilized in these experiments; no preconditioning is used. Now, we note that this linear solver is *Matrix-Free*, meaning that the matrix $J(\mathbf{u}^k)$ is not explicitly required, only its *action* on a vector is needed which translates to significant storage benefits. Unfortunately, due to the complexity of the SIME, even formulating the action of the Jacobian will be both computationally intensive and prone to errors. Thus, previous implementations [19] of this method approximate the Jacobian's action via a first order Gateaux derivative approximation, making this method fully *Jacobian-Free*, i.e. they use [19]

$$J(\mathbf{u}^k) \mathbf{v} \approx \frac{\mathbf{F}(\mathbf{u}^k + \epsilon \mathbf{v}) - \mathbf{F}(\mathbf{u}^k)}{\epsilon}, \quad (3.25)$$

where ϵ is a small perturbation and the error is $O(\epsilon)$. Eq. (3.25) is analogous to a forward difference approach. Consequently, using a centered differences,

$$J(\mathbf{u}^k)\mathbf{v} \approx \frac{\mathbf{F}(\mathbf{u}^k + \epsilon\mathbf{v}) - \mathbf{F}(\mathbf{u}^k - \epsilon\mathbf{v})}{2\epsilon}, \quad (3.26)$$

one can achieve a second order approximation, $O(\epsilon^2)$, for the Gateaux derivative. Now although (3.26) is already a better approximation than (3.25), we take it a step further by expanding (3.26) according to (3.22), which after some re-arranging results in

$$\begin{aligned} \frac{\mathbf{F}(\mathbf{u}^k + \epsilon\mathbf{v}) - \mathbf{F}(\mathbf{u}^k - \epsilon\mathbf{v})}{2\epsilon} &= \frac{A(\mathbf{u}^k + \epsilon\mathbf{v})\mathbf{u}^k - A(\mathbf{u}^k - \epsilon\mathbf{v})\mathbf{u}^k}{2\epsilon} + \\ &\quad \frac{A(\mathbf{u}^k + \epsilon\mathbf{v})\epsilon\mathbf{v} + A(\mathbf{u}^k - \epsilon\mathbf{v})\epsilon\mathbf{v}}{2\epsilon} \\ &- \frac{\mathbf{b}(\mathbf{u}^k + \epsilon\mathbf{v}) - \mathbf{b}(\mathbf{u}^k - \epsilon\mathbf{v})}{2\epsilon}. \end{aligned} \quad (3.27)$$

Then taking the limit in the last two terms as $\epsilon \rightarrow 0$, we get the final form of our Jacobian approximation,

$$\begin{aligned} J(\mathbf{u}^k)\mathbf{v} &\approx \frac{A(\mathbf{u}^k + \epsilon\mathbf{v})\mathbf{u}^k - A(\mathbf{u}^k - \epsilon\mathbf{v})\mathbf{u}^k}{2\epsilon} \\ &+ A(\mathbf{u}^k)\mathbf{v} - D\mathbf{b}(\mathbf{u}^k)\mathbf{v}, \end{aligned} \quad (3.28)$$

where the third term represents the contribution from the Jacobian Matrix of $\mathbf{b}(\mathbf{u}^k)$, which is easily calculated in a closed form, the second term is the contribution from the linear portion of the Jacobian Matrix of $A(\mathbf{u}^k)\mathbf{u}^k$ and the first term is a second order approximation of the contribution from the nonlinear part. Thus, this method amounts to computing the “simple” parts of the Jacobian in closed form and using a second order Gateaux derivative approximation for the remaining terms.

With the above methods in mind, the final step is to provide stopping criteria for both the nonlinear and linear solvers. For the linear solver, we follow [19] and set

$$\|J(\mathbf{u}^k)\tilde{\mathbf{u}}^k + \mathbf{F}(\mathbf{u}^k)\| < \gamma(k)\|\mathbf{F}(\mathbf{u}^k)\|, \quad (3.29)$$

as our convergence criterion for the GMRES solver. Here $\tilde{\mathbf{u}}^k$ is the approximate solution to (3.24), $\|\cdot\|$ represents the L_2 functional norm, and $\gamma(k)$ is given by [19]

$$\gamma(k) = \begin{cases} \gamma_{ini} & \text{if } \|\mathbf{F}(\mathbf{u}^k)\| \geq res_t \\ \min(\gamma_{ini}, \|\mathbf{F}(\mathbf{u}^k)\|/\|\mathbf{F}(\mathbf{u}^{k-1})\|) & \text{if } \|\mathbf{F}(\mathbf{u}^k)\| < res_t, \end{cases} \quad (3.30)$$

where $\gamma_{ini} = 0.99$. This approach is referred to as an “inexact” Newton Method; in [22] it is noted that over solving of (3.24) in early iterations can not only be detrimental to computational costs, but it can also prevent the method to converge at all! It should be noted that in [22] and [19], they vary the value of res_t with spatial resolution but for this study we keep it constant at 0.625; studying the effects of the value is left for future work.

To create a stopping criterion for the Newton iterations, we propose using a tolerance proportional to the discretization error. Unfortunately, as we are working in a dimensionalized system, careful consideration must be taken to match the units of the discretized equation. Using the Coriolis parameter f to set a reference time scale, we set our nonlinear stopping criterion to

$$\|\mathbf{F}(\mathbf{u}^k)\| < \rho h_0 f u_0 \epsilon_{tol}, \quad (3.31)$$

where u_0 is a typical sea ice velocity, chosen to be 0.1 m/s, $h_0 = 1$ m is a reference ice thickness, the ice density is set to $\rho = 900$ kg m⁻³, and

$$\epsilon_{tol} = \gamma_{nl} \left(\frac{dx}{L_x} \right)^2, \quad (3.32)$$

where L_x is the length scale of our domain, chosen to be the domain extent in the x -direction, and γ_{nl} is a tolerance coefficient

parameter whose default value is $\gamma_{nl} = 10$. As it will be demonstrated in Section 4.5, the numerical solution can be sensitive to the choice of the value of γ_{nl} , and its optimal value remains resolution dependent. A universal expression for ϵ_{tol} remains to be discovered.

4. Numerical tests and validation

To validate and assess our solver’s performance, we elect to test its ability to reproduce a known solution. Unfortunately, the SIME does not have any known analytical solutions. To get around this, we follow [29] and construct our own solution. It is always possible to do so if appropriate forcing is added to the governing equations, the only requirement is that the chosen solution has enough structure to exercise higher-derivative calculations [30]. The solution need not be physical in any way. If we define

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix},$$

to be our chosen solution, this procedure amounts to solving

$$\begin{aligned} -\rho h \frac{\partial \mathbf{u}}{\partial t} - \rho h f (\mathbf{k} \times \mathbf{u}) + \tau_a - \tau_w + \nabla \cdot \boldsymbol{\sigma} - \rho h g \nabla H_d \\ = \mathcal{L}(\mathbf{w}), \end{aligned} \quad (4.1)$$

where $\mathcal{L}(\mathbf{w})$ is a known forcing term at each grid point, defined by

$$\begin{aligned} \mathcal{L}(\mathbf{w}) &= -\rho h \frac{\partial \mathbf{w}}{\partial t} - \rho h f (\mathbf{k} \times \mathbf{w}) + \tau_a - \tau_w(\mathbf{w}) + \nabla \cdot \boldsymbol{\sigma}(\mathbf{w}) \\ &- \rho h g \nabla H_d, \end{aligned} \quad (4.2)$$

which can be calculated exactly for any known \mathbf{w} .

For our known solution, we choose the two dimensional, moving sine wave,

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{10} \sin \left[\left(\frac{4x}{L_x} - 2 \right)^2 + \left(\frac{4y}{L_y} - 2 \right)^2 + ct \right] \\ \frac{1}{10} \cos \left[\left(\frac{4x}{L_x} - 2 \right)^2 + \left(\frac{4y}{L_y} - 2 \right)^2 + ct \right] \end{bmatrix}, \quad (4.3)$$

where $L_x = L_y = 2000$ km, are our domain’s extent in the x - and y -directions, and $c = 5 \times 10^{-6}$ s⁻¹ is the phase. In addition to this solution having significant structure it also has values of similar order to that expected for sea ice velocities. Fig. 4.1 shows this solution with $t = 0$ s, over the entire domain.

For the actual experiment, we limit the computational domain to two separate ice patches in, respectively, the south-west and north-east corners of a square ocean basin, surrounded by land, as shown in a toy example in Fig. 4.2. The solution over the restricted domain is shown in Fig. 4.3. It has to be noted that special care has to be taken when building the nonlinear system in (3.21) so that this limitation of the computational domain to ice only areas can be adopted to arbitrary geometries.

The reasoning behind this domain choice is two fold. First, it allows us to test all boundary types, i.e. open water boundaries to the north, east, south, and west, and likewise with land boundaries. Second, note the local minima and maxima of the test solution near the center of the domain in Fig. 4.1; the presence of these extrema will lead to a capping of the viscosities which may cause large errors and plague our convergence estimates. Thus, for the sake of convergence tests, we avoid this area.

It should be noted that in order to properly produce (4.3), slight modifications must be made to the boundary conditions. We still use Dirichlet conditions at land boundaries and Neumann conditions at open water boundaries, but they are no longer homogeneous, i.e. we use

$$\mathbf{u} = \beta(\mathbf{x}, t) \quad \text{at land boundaries}, \quad (4.4)$$

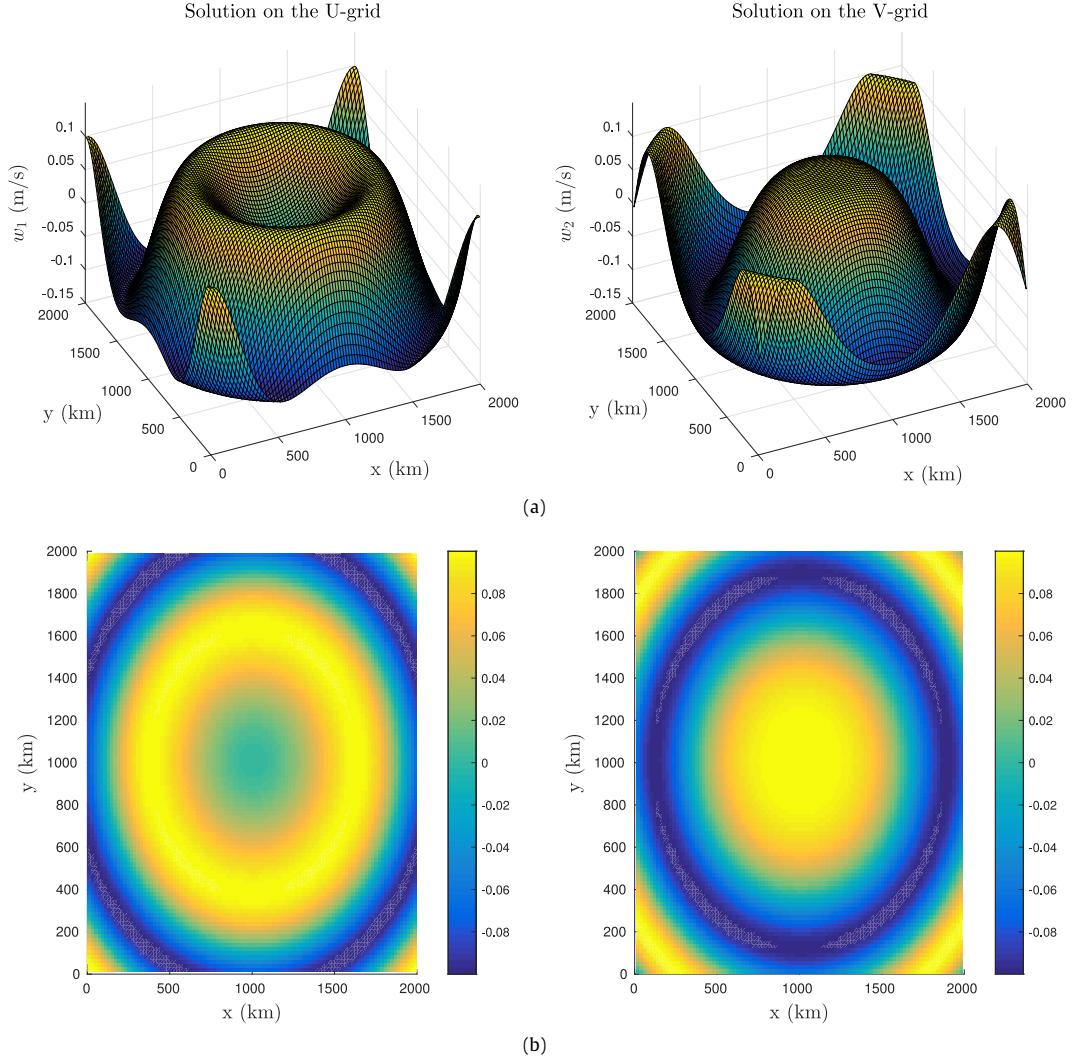


Fig. 4.1. Structure of the test function over the entire square domain, showing w_1 on the U-grid (left) and w_2 on the V-grid (right).

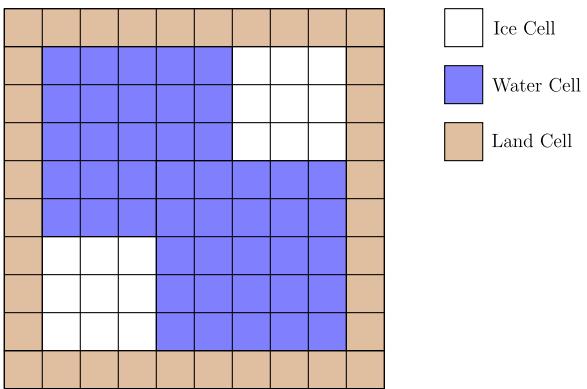


Fig. 4.2. Illustration of the computational setup, showing the computational domain extent, occupied by two ice patches located at two opposite corners of the ocean basin, surrounded by land and a mock grid mesh overlaid.

and

$$\frac{\partial \mathbf{u}}{\partial \mathbf{n}} = \alpha(\mathbf{x}, t) \quad \text{at open water boundaries}, \quad (4.5)$$

where β and α are the known solution and its derivatives determined from (4.3). Once a solver is given the ability to use non-homogeneous conditions, homogeneous conditions can be easily applied as they are only a special case of (4.4) and (4.5) with $\beta(\mathbf{x}, t) = \alpha(\mathbf{x}, t) = 0$.

4.1. Validation simulation

In current ESMs, solvers are typically ran on time scales on the order of months to years. These scales are not practical for validation tests, thus we elect to test the validation problem over a week long simulation. This temporal scale allows for effective numerical testing yet provides preliminary insight into longer time scales. For all simulations, we set our physical parameters as laid out in [19]. Table 4.1 contains the chosen numerical parameters for this validation simulation.

Although it should have little to no affect on the validation simulation, a forcing field must be specified. We follow [21] and set \mathbf{u}_w^g and \mathbf{u}_a^g according to

$$\mathbf{u}_w^g = \begin{bmatrix} u_w^g \\ v_w^g \end{bmatrix} = \begin{bmatrix} 0.1(2y - L_y)/L_y \\ -0.1(2x - L_x)/L_x \end{bmatrix}, \quad (4.6)$$

and

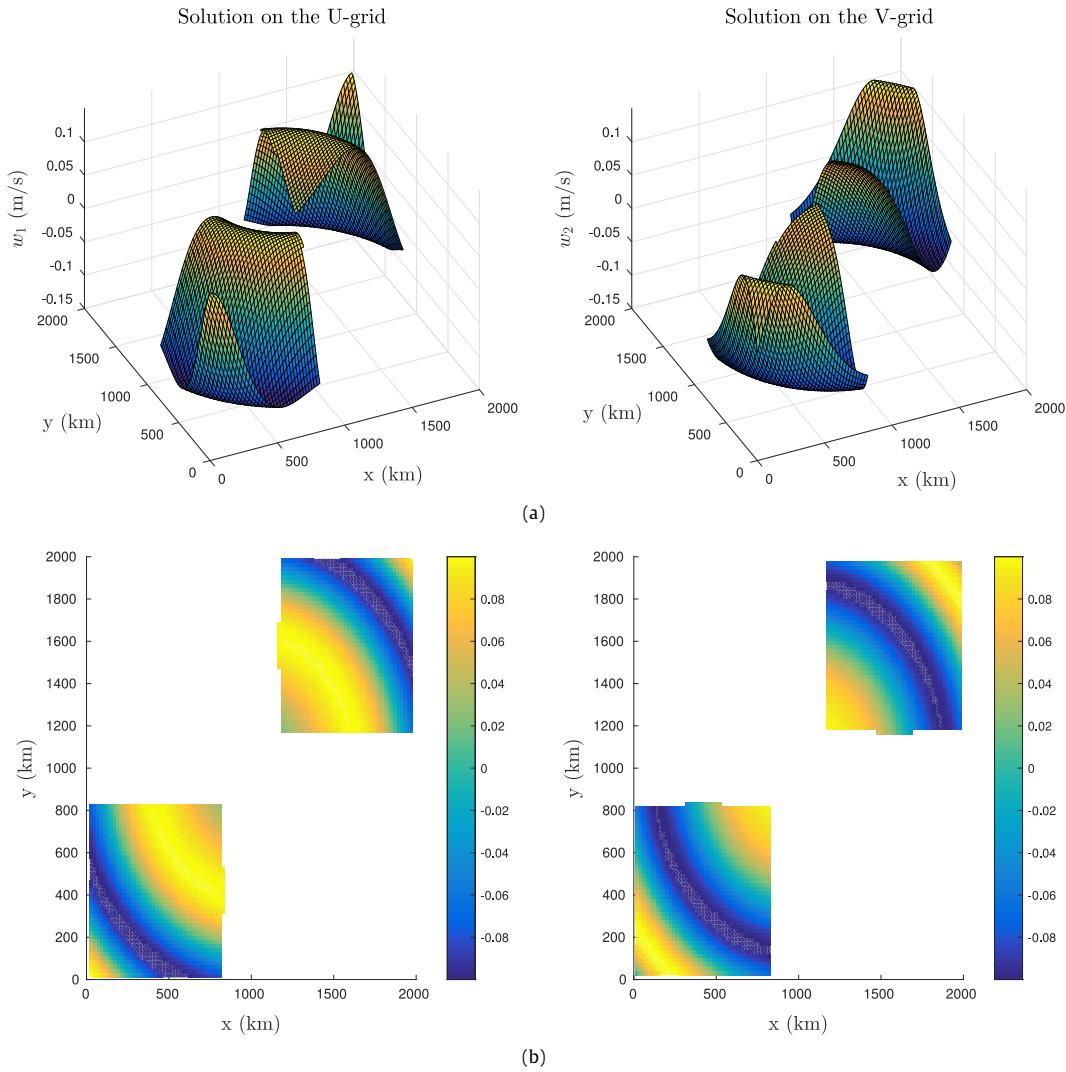


Fig. 4.3. Validation test solution restricted to the ice-only computational domain, showing w_1 on the U-grid (left) and w_2 on the V-grid (right) for the limited domain.

Table 4.1

Numerical parameters for validation simulation.

Symbol	Definition	Value
dx	Spatial resolution	20 (km)
dt	Temporal resolution	10 (min)
L_x	X-extent	2000 (km)
L_y	Y-extent	2000 (km)
T	Final time	7 (days)
ϵ	Jacobian perturbation	10^{-6}
res_t	Residual transition	0.625
γ_{nl}	Nonlinear tolerance coef.	10
k_{max}	Maximum number of nonlinear iterations	200

$$\begin{aligned} \mathbf{u}_a^g &= \begin{bmatrix} u_a^g \\ v_a^g \end{bmatrix} \\ &= \begin{bmatrix} 5 + \left[\sin\left(\frac{2\pi t}{\Theta}\right) - 3 \right] \sin\left(\frac{2\pi x}{L_x}\right) \sin\left(\frac{\pi y}{L_y}\right) \\ 5 + \left[\sin\left(\frac{2\pi t}{\Theta}\right) - 3 \right] \sin\left(\frac{\pi x}{L_x}\right) \sin\left(\frac{2\pi y}{L_y}\right) \end{bmatrix}, \quad (4.7) \end{aligned}$$

where Θ is the wind forcing period, set to 4 days. Note that if both the solver and validation forcing are behaving as planned, the effects of (4.7) should be canceled completely as the air drag term is independent of the actual solution.

Table 4.2

Validation simulation errors, $\mathbf{E} = \text{abs}(\mathbf{u}_{\text{exact}} - \mathbf{u})$.

t (Days)	U-grid		V-grid	
	$\ \mathbf{E}\ _{L_2}$	$\ \mathbf{E}\ _{L_\infty}$	$\ \mathbf{E}\ _{L_2}$	$\ \mathbf{E}\ _{L_\infty}$
1	1.04e-04	8.50e-04	8.64e-05	9.37e-04
2	1.02e-04	6.86e-04	6.41e-05	3.68e-04
3	1.00e-04	6.47e-04	6.59e-05	3.78e-04
4	9.48e-05	6.74e-04	6.19e-05	3.57e-04
5	1.13e-04	2.08e-03	6.13e-05	9.56e-04
6	1.04e-04	7.20e-04	5.99e-05	3.70e-04
7	1.89e-04	2.74e-03	1.87e-04	3.15e-03

Once ran, the solver behaved quite well. At this temporal resolution, progressing 7 days amounts to 1008 time steps and not once did the solver fail to converge. In fact, as shown in Fig. 4.4 the solver never got close to the maximum allowable iterations of 200; the mean iteration count to reach convergence was 8.40, with a standard deviation of 0.93.

The solver was set to save the resulting solution and residual progress at the end of every day. Table 4.2 contains the resulting error norms at these benchmark time levels and as an example, Fig. 4.5 shows the absolute error at the end of the simulation and the residual progression for the final solve, which represents the typical behavior seen throughout the simulation. Although there are some larger errors near the center of the domain in Fig. 4.5a,

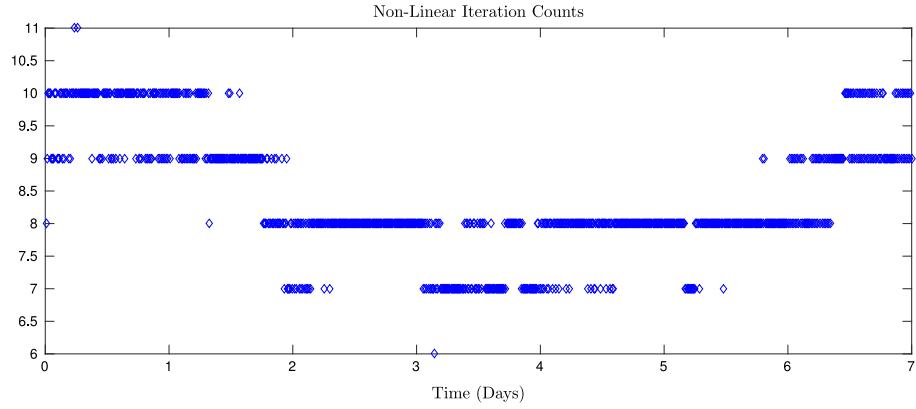


Fig. 4.4. Number of nonlinear iterations until convergence per time step along the duration of the validation simulation.

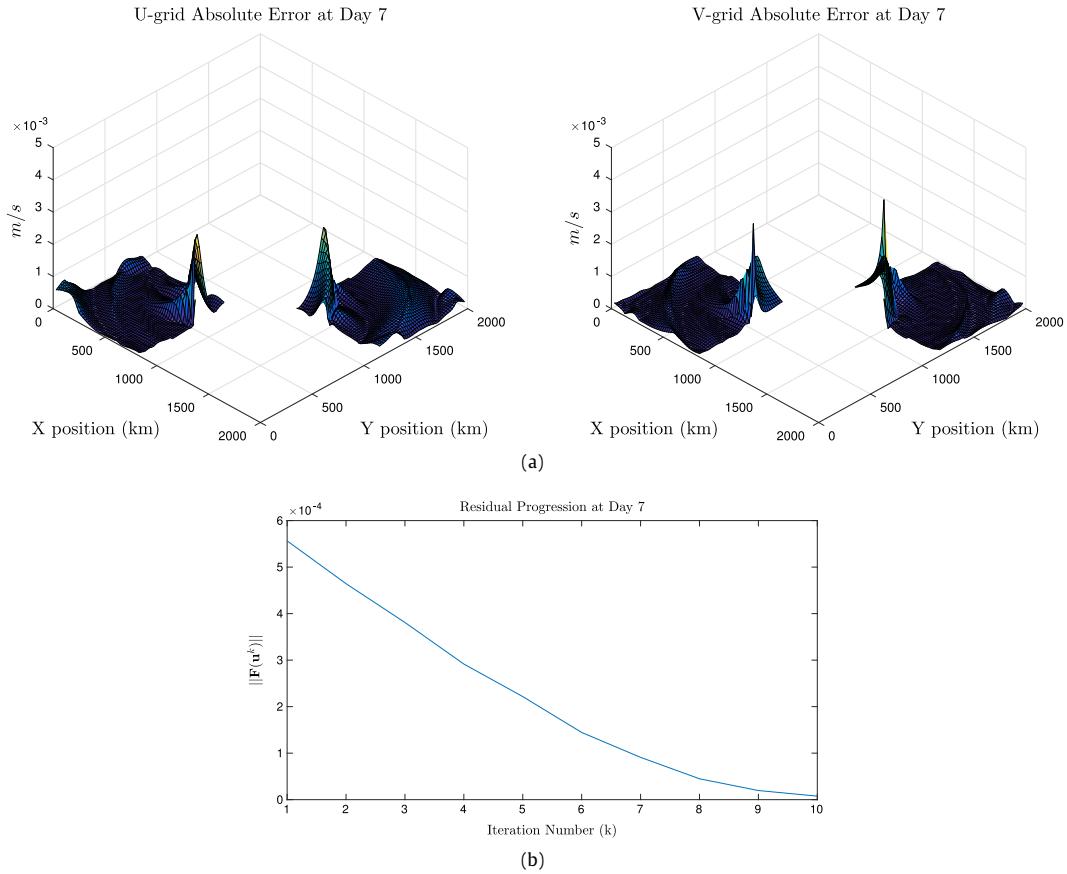


Fig. 4.5. (a) Absolute error at the end of day 7 for the validation simulation. (b) Typical progression of nonlinear residual over each solve (final solve shown).

they are relatively unnoticeable in the solution. This is easily confirmed by an inspection of the additional numerical solutions and their associated errors, provided separately for conciseness in [24]. Additionally, as seen in Table 4.2 the error measures for day 7 are the highest throughout the entire simulation, as one would expect because of error accumulation. However, according to Table 4.2, a major error increase is seen only at day 7, interestingly the errors were stable and even slightly decreasing with time during the first 6 days.

It is worth noting that the theoretical convergence rate for Newton solvers is quadratic, but as seen in Fig. 4.5b, we do not see

this theoretical rate. Like [22], this can likely be attributed to the fact that an inexact Newton method is used and thus there is some error in the calculation of the Newton iterate. As stated in [22], it is possible that asymptotic quadratic convergence could be attained with a small value of $\gamma(k)$ in (3.30), but further investigation is required. Another possible contributor to the lack of quadratic convergence is an instability problem inherent to the VP model, which may not be damped by the Crank–Nicolson scheme and therefore plague the residual progression; further discussion on this topic is provided in the conclusion section of this paper. Nevertheless, this solver has been shown to consistently produce accurate solutions

in a reasonable number of nonlinear iterations and its convergence rate has been deemed acceptable.

4.2. Grid-Refinement tests

With the successful simulation described above, we now move on to convergence tests. Due to our fully second order discretization, it is expected that in the limiting behavior as $dx, dt \rightarrow 0$, we will see second order convergence. To assess this, we perform grid-refinement tests and run the validation simulation with three different resolutions:

- *Coarse Resolution*: $dx|dt = 40 \text{ km}|20 \text{ min}$
- *Moderate Resolution*: $dx|dt = 20 \text{ km}|10 \text{ min}$
- *Fine Resolution*: $dx|dt = 10 \text{ km}|5 \text{ min}$.

With these simulations, the convergence rates were estimated between the coarse and moderate resolutions and the moderate and fine resolutions; Tables 4.3 and 4.4 show the resulting error norms and convergence rate estimates, p , calculated via

$$p(dx, dt) = \log \left(\frac{\|\mathbf{E}(2dx, 2dt)\|}{\|\mathbf{E}(dx, dt)\|} \right) / \log(2).$$

The error was calculated using both the L_2 and L_∞ norms to assess both global and point wise convergence, respectively. We note that these are the max and energy norms and are taken with respect to the spacial variables (x, y) with the time t kept fixed as indicated in the table. Also, we recall that in the infinite dimensional functional space of solutions, these two norms are not equivalent.

The second order convergence is seen in both the L_2 and L_∞ norms for four of the six days measured, in particular when going from the moderate to fine resolution. As the theoretical convergence rate is only expected in the limiting behavior as $dx, dt \rightarrow 0$, it is not surprising that it is not observed every where, and thus these results are accepted as validating the second order convergence when both dx and dt go to zero. This is in fact typical for highly complex (nonlinear) equations where an under performing rate of convergence ($p < 2$), when moving from one grid to the next in the refinement hierarchy, is compensated with an over performance ($p > 2$) at the next or at the previous grid refinement level. This is mainly due to the fact that, at coarse resolutions, the complexity of the solution may be changing with the changing grid. This is more often the case, even at fine resolutions, when the equations develop instabilities at small scales, which is the case for the SIME model [14] as discussed in the conclusion section.

Now, although the solver performance seems to be consistent with the theoretical convergence rate, there are still some problems. For instance, beyond the first day, the convergence rates between the coarse and moderate resolutions are significantly higher than the theoretical expectation. This has been attributed to convergence problems noted for the coarse simulation. As shown in Fig. 4.6a, the coarse resolution simulation had trouble converging at various times throughout the simulation, where many more than the standard sub 10 iterations were required; Fig. 4.6b shows a specific example of one of these solves. When this occurred, it appears that the solver was close to the actual solution in the first few iterations but for unknown reasons, was unable to get below the specified tolerance. The solver then appears to search around for a solution, at some points making large steps away from it, until coming back to the solution neighborhood and converging. Note that although for this specific case, convergence was reached, this is not always so. It is believed that this “searching” behavior deteriorates the solution and introduces the added error seen in the coarse simulation. This added error inflates the difference between the coarse and moderate simulations, plaguing the convergence estimates.

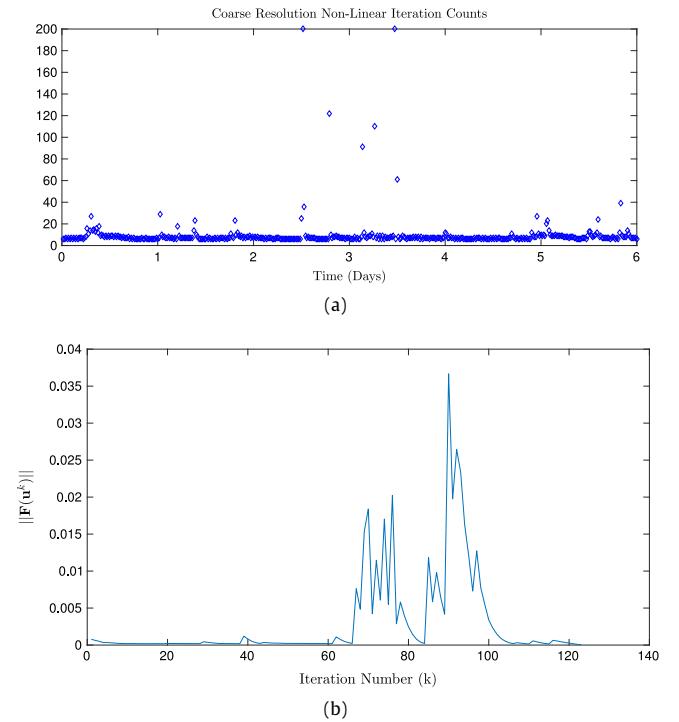


Fig. 4.6. Coarse simulation convergence problems: (a) Iteration counts for the coarse resolution simulation. (b) Residual progression at $t = 2.8$ days.

It should be noted that convergence problems were not limited to the coarse simulation. Although the fine simulation behaved very well over the first six days, taking ~ 10 iterations for every solve and producing acceptable errors, after the sixth day the solver broke down and stopped converging altogether, which is why day 7 is not included in Tables 4.3 and 4.4. The specific reason for this is unknown, but this problem and the issues with the coarse simulation point towards a resolution sensitivity that should be explored in future work. The occasional convergence failure of the JFNK method for the SIME was also reported in [22]. It is possible that this is due to inherent small scale instability of the VP equations, reported in the literature [14], as discussed in the conclusion section.

4.3. The Jacobian approximation: failure of the first order scheme

In addition to assessing the convergence rate of this solver, we wish to assess the effects of using the second order approximation for the Jacobian multiplication in Eq. (3.28). To determine if this second order approximation is not overkill, we re-ran the moderate resolution simulation with the following first order approximation,

$$J(\mathbf{u}^k)\mathbf{v} \approx \frac{A(\mathbf{u}^k + \epsilon\mathbf{v})\mathbf{u}^k - A(\mathbf{u}^k)\mathbf{u}^k}{\epsilon} + A(\mathbf{u}^k)\mathbf{v} - D\mathbf{b}(\mathbf{u}^k)\mathbf{v}. \quad (4.8)$$

This simulation utilized the same parameters as those defined in Table 4.1 but due to convergence problems, the first order simulation only progressed two hours or twelve time steps; solutions were saved every 20 min. Fig. 4.7a shows a comparison of the resulting iteration counts between the second and first order simulations, while Fig. 4.7b shows comparisons of the resulting error norms.

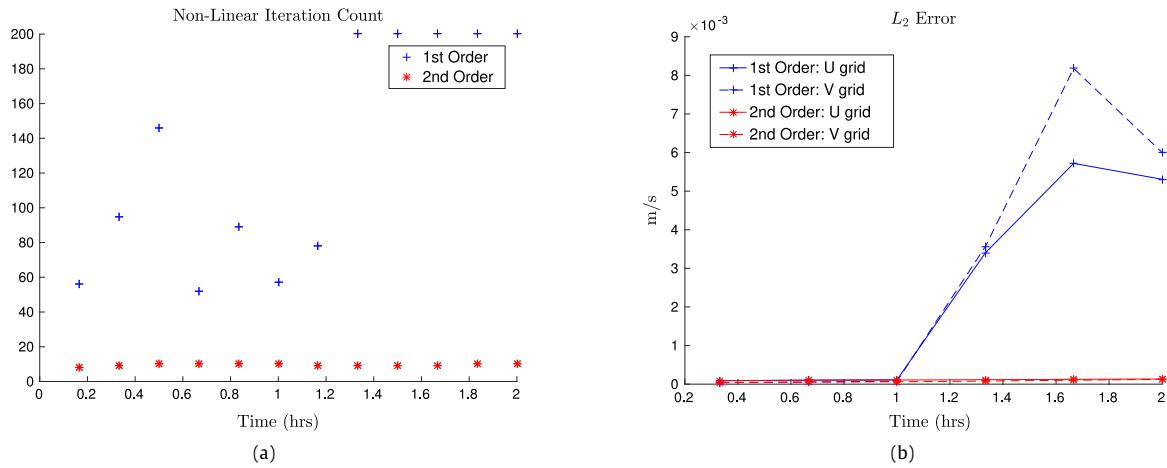
Prior to the first order simulation failing at the one hour and twenty minute mark, it produced comparable errors to those from its second order counterpart. Regardless, as shown in Fig. 4.7a, to produce these solutions the first order solver used significantly

Table 4.3Estimated convergence rates (p) in the L_2 norm.

dx/dt (km)/(min)	Day 1		Day 2		Day 3		Day 4		Day 5		Day 6	
	$\ \mathbf{E}\ _{L_2}$	p	$\ \mathbf{E}\ _{L_2}$	p	$\ \mathbf{E}\ _{L_2}$	p	$\ \mathbf{E}\ _{L_2}$	p	$\ \mathbf{E}\ _{L_2}$	p	$\ \mathbf{E}\ _{L_2}$	p
U-grid convergence												
40/20	4.02e–04	–	1.78e–03	–	1.11e–03	–	2.50e–03	–	1.92e–03	–	1.14e–03	–
20/10	1.04e–04	2.0	1.02e–04	4.1	1.00e–04	3.5	9.48e–05	4.7	1.13e–04	4.1	1.04e–04	3.5
10/5	7.26e–05	0.5	2.58e–05	2.0	2.45e–05	2.0	2.35e–05	2.0	8.95e–05	0.3	2.53e–05	2.0
V-grid convergence												
40/20	2.90e–04	–	1.68e–03	–	8.52e–04	–	1.55e–03	–	1.75e–03	–	1.08e–03	–
20/10	8.64e–05	1.7	6.41e–05	4.7	6.59e–05	3.7	6.19e–05	4.6	6.13e–05	4.8	5.99e–05	4.2
10/5	1.89e–04	–1.1	1.62e–05	2.0	1.64e–05	2.0	1.53e–05	2.0	5.29e–05	0.2	1.39e–05	2.1

Table 4.4Estimated convergence rates (p) in the L_∞ norm.

dx/dt (km)/(min)	Day 1		Day 2		Day 3		Day 4		Day 5		Day 6	
	$\ \mathbf{E}\ _{L_\infty}$	p	$\ \mathbf{E}\ _{L_\infty}$	p	$\ \mathbf{E}\ _{L_\infty}$	p	$\ \mathbf{E}\ _{L_\infty}$	p	$\ \mathbf{E}\ _{L_\infty}$	p	$\ \mathbf{E}\ _{L_\infty}$	p
U-grid convergence												
40/20	3.46e–03	–	2.85e–02	–	2.03e–02	–	4.70e–02	–	2.37e–02	–	1.61e–02	–
20/10	8.50e–04	2.0	6.86e–04	5.4	6.47e–04	5.0	6.74e–04	6.1	2.08e–03	3.5	7.20e–04	4.5
10/5	6.91e–04	0.3	1.65e–04	2.1	1.57e–04	2.0	1.47e–04	2.2	4.79e–03	–1.2	2.04e–04	1.8
V-grid convergence												
40/20	4.33e–03	–	2.99e–02	–	1.65e–02	–	2.39e–02	–	2.91e–02	–	2.87e–02	–
20/10	9.37e–04	2.2	3.68e–04	6.3	3.78e–04	5.5	3.57e–04	6.1	9.56e–04	4.9	3.70e–04	6.3
10/5	2.82e–03	–1.6	9.45e–05	2.0	1.06e–04	1.8	8.83e–05	2.0	2.41e–03	–1.3	7.98e–05	2.2

**Fig. 4.7.** Comparison of solver performance with first and second order Jacobian approximations: (a) Iteration count (b) L_2 error.

more computational resources and reached the solution through a rather spurious path (Fig. 4.8a). Tests on the linear solver alone (not shown here) suggest that, using the first order approximation, there are some issues when solving the linear system at the first nonlinear iteration; these troubles are likely to blame for the large spike in the residual seen at the first iteration in Fig. 4.8a. As reported in Lemieux et al. [22], the large residuals are often localized over a few grid points only, so a smoothing technique of some sort may be adopted to help alleviate the problem. After the one hour and twenty minute mark, the first order simulation breaks down completely and is no longer capable of converging in 200 iterations and introduces the large errors seen in Figs. 4.7b and 4.8b. Due to the significant reduction in computational resources and improvement in convergence behavior, the second order approximation constitutes a drastic improvement for the JFNK approach for the SIME.

4.4. Conditional termination

Motivated by the searching behavior seen in the coarse simulation, we additionally propose to use a “conditional termination”

to limit the computational effort. For this solver to work, between time steps, the solution cannot undergo drastic changes. Therefore, as the previous time step’s solution is used as the initial iterate for the nonlinear solver, it should be “close” to the solution in the early iterates and large steps away are likely detrimental. With this in mind, conditional termination adopts a “good enough” approach and, given a tolerance $r > 1$, when going from iteration k to $k + 1$, if

$$\|\mathbf{F}(\mathbf{u}^{k+1})\| > r \|\mathbf{F}(\mathbf{u}^k)\|$$

reject \mathbf{u}^{k+1} and accept \mathbf{u}^k as the solution, terminating the solver.

To test this method, we set $r = 2$ and re-ran the coarse simulation, as this was the only resolution that experienced this searching behavior. We then compared the solutions produced via conditional termination and those produced via the original simulation, referred to as the control; Fig. 4.9 shows the resulting comparison. It can be seen that for all days considered, the conditional termination run produces comparable errors to those from the control simulation and in most cases there is actually a reduction.

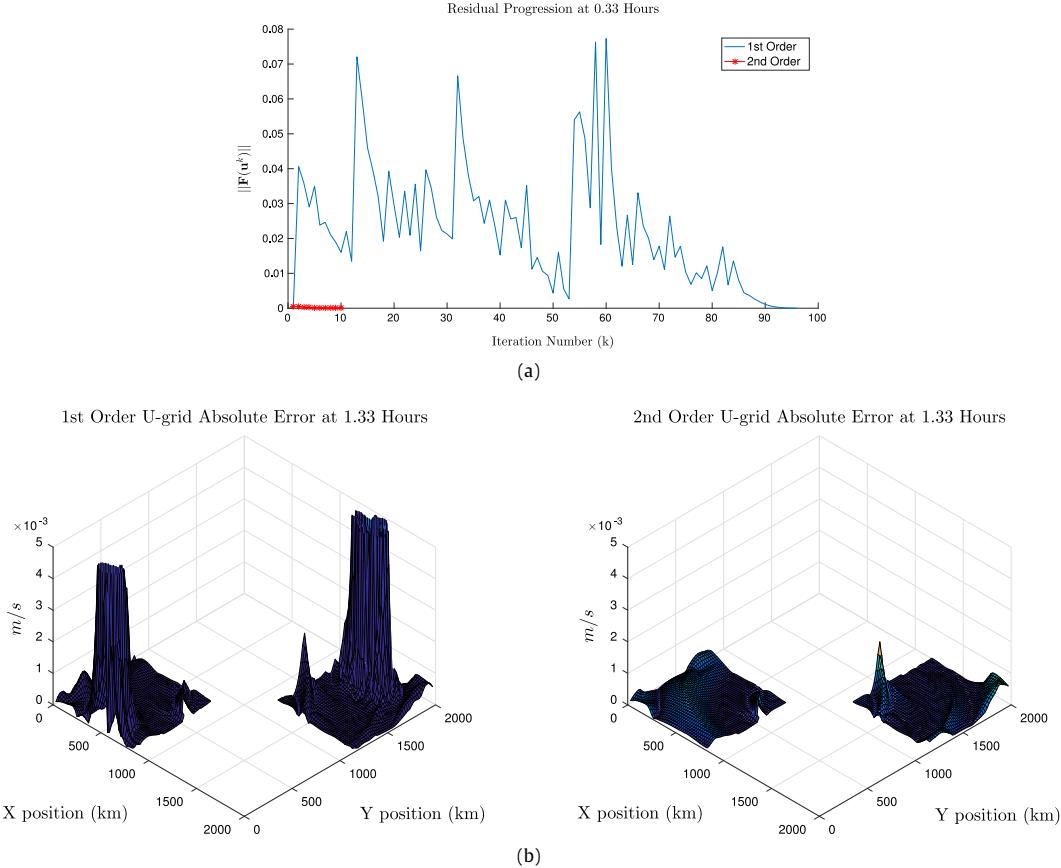


Fig. 4.8. First order Jacobian solver convergence problems: (a) Typical convergence behavior. (b) Absolute error after first order Jacobian solver breaks down.

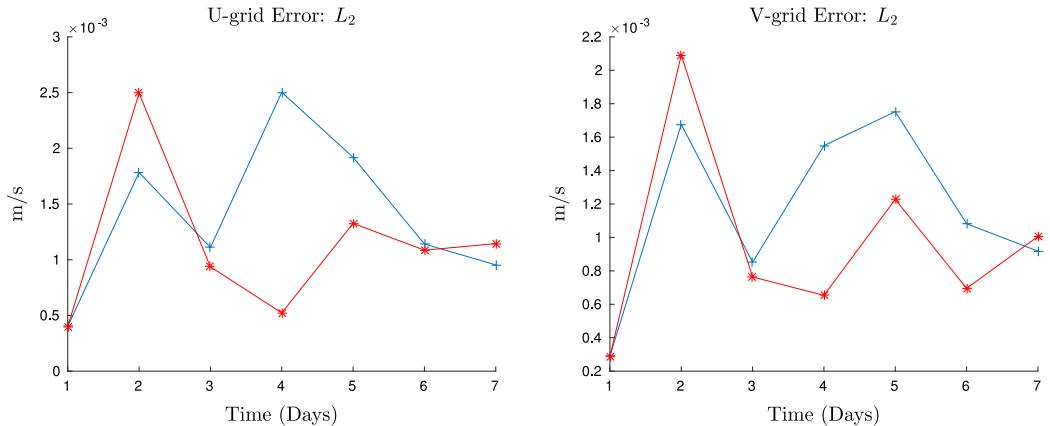


Fig. 4.9. Conditional termination (red, stars) vs. control (blue, crosses): absolute error comparison. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

As this method does not significantly alter the error, the main benefit comes from saving computational effort. As discussed in Section 4.2, the solver sometimes steps away from the solution neighborhood, searching sporadically until it eventually returns. This sporadic search uses valuable computational resources without the pay off of reduced error and can even deteriorate the solution, which is manifested in failed convergence and larger errors. The conditional termination saves this wasted effort by not allowing the solver to leave the solution neighborhood. As can be seen in Fig. 4.10, this leads to a significant reduction in the number of nonlinear iterations when the solver has problems

converging. In the context of ESMs, saving computational effort is a considerable benefit and thus this method should be considered and tested in the context of actual geophysical simulations.

4.5. Sensitivity to ad-hoc parameters

For our final numerical tests, we wish to assess the impact of ad-hoc parameters, specifically the nonlinear tolerance coefficient, γ_{nl} , and the small perturbation, ϵ , used in the approximation to the Jacobian multiplication.

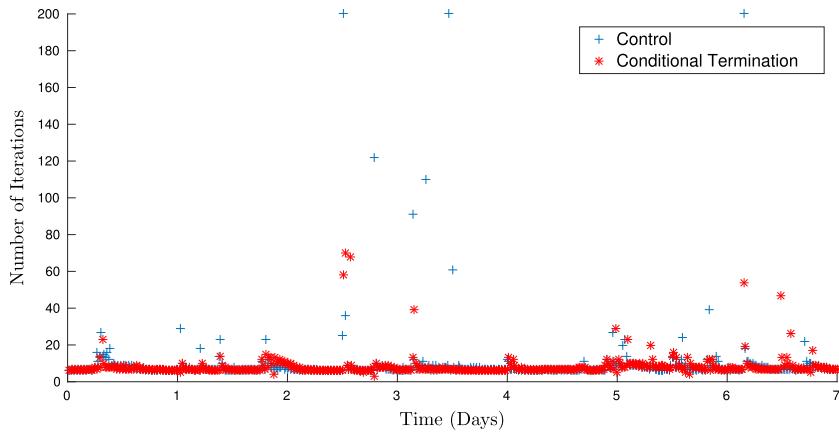


Fig. 4.10. Conditional termination vs. control: nonlinear iteration counts.

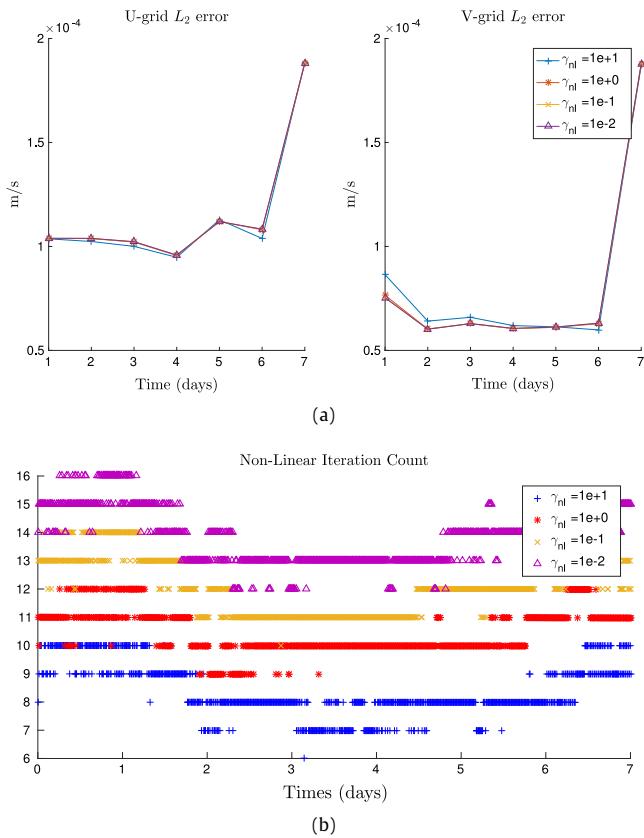


Fig. 4.11. Tighter nonlinear tolerance tests: (a) L_2 error at the end of each day. (b) Iteration counts over the entire simulation.

4.5.1. Nonlinear tolerance coefficient: γ_{nl}

For the primary validation simulation, γ_{nl} was set to 10. To see if added accuracy could be gained by decreasing this value, we re-ran the simulation (at the moderate resolution) with $\gamma_{nl} = 1, 0.1$, and 0.01 ; Fig. 4.11a shows the resulting L_2 errors at the end of each day and Fig. 4.11b shows iteration counts over the entire simulation.

As can be seen from Fig. 4.11, the solver still performed well with these tighter tolerances, consistently converging within 20 iterations. Nevertheless, to produce the lower residuals, additional iterations were required and no considerable change in error was noted. Thus, these tighter tolerances only result in wasted computational effort.

To see if added computational effort could be saved with $\gamma_{nl} > 10$, an additional run was attempted with $\gamma_{nl} = 100$ but the solver broke down at the 3 h mark (18 time steps), consistently failing to converge in the allowable iteration count. This suggests that, for the validation simulation, at the moderate resolution, a choice of $\gamma_{nl} \sim 10$ is near the optimal value; setting it smaller only results in wasted computational resources while setting it larger causes the solver to break down. However, it should be noted that the looser tolerance affects each resolution differently. For instance, with $\gamma_{nl} = 100$, the solver is capable of completing the coarse resolution simulation and preliminary tests show little affect on the fine simulation; Fig. 4.12a shows the resulting error comparison for the coarse simulation with $\gamma_{nl} = 10$ and 100, while Fig. 4.12b shows the iteration counts. While the looser tolerance results in similar errors, it significantly reduces the searching behavior in the coarse simulation prior to day 6. This difference in behavior between resolutions suggests that additional consideration is required in the derivation of the convergence formula (3.31); further study is required but perhaps a linear formula is more suitable.

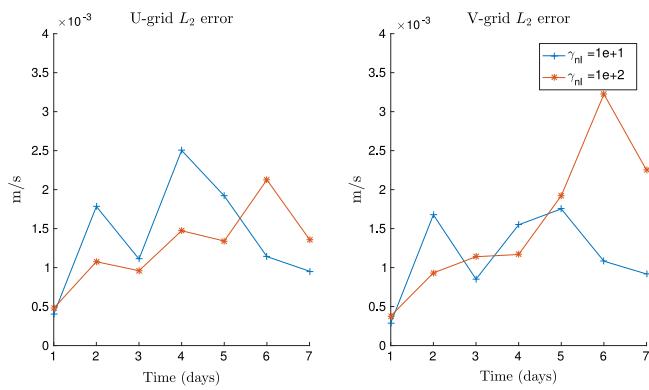
4.5.2. Jacobian approximation perturbation: ϵ

For our study, we followed [19] and set $\epsilon = 10^{-6}$ as our default value. To assess the effect of this parameter, we re-ran the moderate resolution simulation with $\epsilon = 10^{-4}, 10^{-5}, 10^{-7}$, and 10^{-8} . Fig. 4.13a shows the resulting error norms and Fig. 4.13b shows the iteration counts. As can be seen, changing the value of ϵ in the Jacobian approximation results in negligible changes in both the errors and required nonlinear iterations; there is little sensitivity to its value and thus $\epsilon = 10^{-6}$ is acceptable. In contrast, similar tests conducted with the first order solver (not shown) showed that the choice of ϵ can improve or reduce its ability to converge, further highlighting the benefit of the second order approximation.

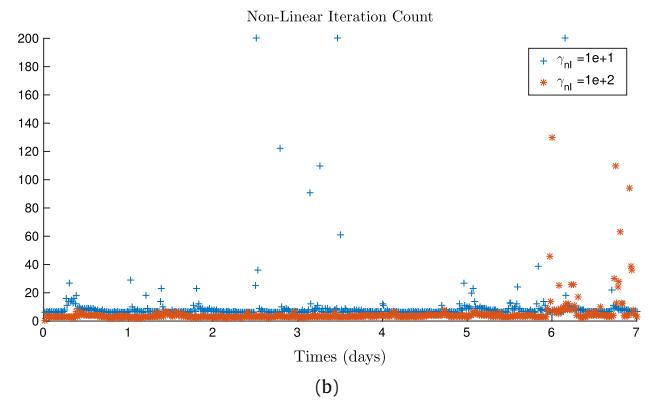
5. Conclusion

Sea ice dynamics plays a central role in the evolution of the distribution of ice thickness and ice area coverage [6]. The movement of ice floes is subject to a variety of forces including the wind and ocean current stresses acting, respectively, on the top and bottom surfaces of the floe, the tilting of the ocean surface on which the ice floats, the Coriolis force, and internal forces due to its resistance to compression, shear, and tensile stresses [11]. The latter are by far the most complex in terms of mathematical modeling as they involve a highly nonlinear rheology depending on the ice velocity derivatives, in the form of complex constitutive laws.

The most widely accepted constitutive law for ice dynamics, due to Hibler [6,9], takes the form of a viscous-plastic rheology



(a)



(b)

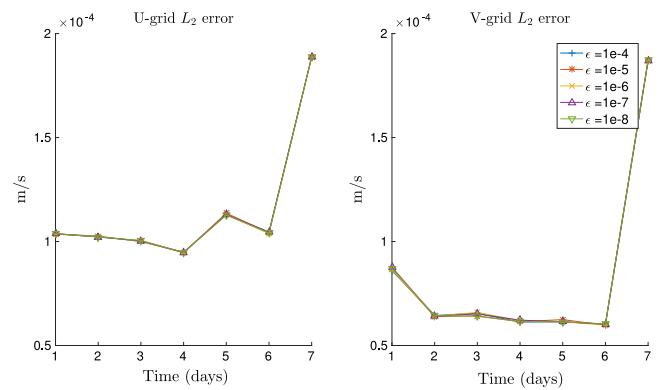
Fig. 4.12. Looser nonlinear tolerance tests with coarse resolution: (a) L_2 error at the end of each day. (b) Iteration counts over the entire simulation.

with an elliptic yield curve. However, its numerical treatment is highly challenging due to the enormous range of effective viscosities in the model [21]. Nonetheless, the relatively recent introduction of a Jacobian free Newton–Krylov numerical scheme by Lemieux et al. [16,19] opened a new opportunity for accurate and efficient methods to solve the sea ice momentum equations (SIME). Here, we build on Lemieux et al.'s work and propose a fully second order numerical scheme for SIME. In addition to using a Crank–Nicolson method instead a backward Euler in time and the use of explicit analytical expressions to assign boundary conditions for the viscosities that are compatible with the prognostic variables, instead of an ad hoc Taylor expansion, we most importantly improved on the approximation of the Jacobian matrix involved in the Newton iterations. Instead of a simple first order approximation of the Gateaux derivative, we propose a more accurate formulation by expressing the linear and nearly linear terms in closed form and use a second order approximation of the Gateaux derivative for the remaining highly nonlinear terms.

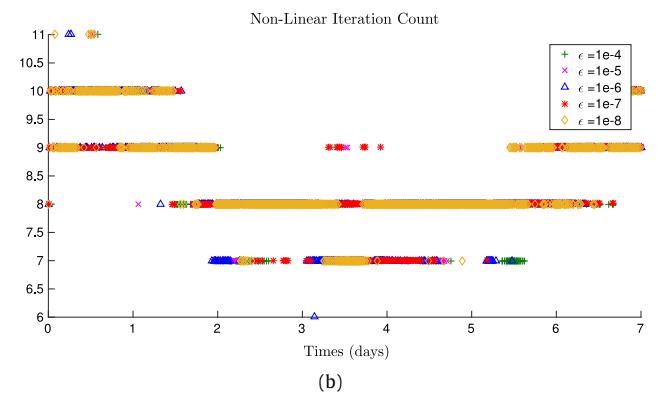
To establish boundary conditions for the viscosities, the rheology terms are expressed explicitly in terms of the derivatives of the ice pressure and velocity components; as shown in Section 3.1, thanks to the hyperbolic tangent smoothing function, introduced by Lemieux et al. [16], the derivatives of ζ and η can be expressed in terms of high order derivatives of velocity components. The use of a Crank–Nicolson scheme allows us to achieve second order accuracy in both temporal and spacial resolution without any significant extra cost since the resulting nonlinearity is precisely the same as in the case of a backward Euler method.

In Section 3.2, the nonlinear system is expressed as

$$F(\mathbf{u}^{n+1}) \equiv A(\mathbf{u}^{n+1})\mathbf{u}^{n+1} - \mathbf{b}(\mathbf{u}^{n+1}) = 0,$$



(a)



(b)

Fig. 4.13. ϵ sensitivity tests: (a) L_2 error at the end of each day. (b) Iteration counts over the entire simulation.

where \mathbf{u}^{n+1} is the solution at the next time step, $n + 1$; the dependence on the solution at time t_n , \mathbf{u}^n , and on the external forcing terms, is not expressed for simplicity in exposition. Here $A(\mathbf{u}^{n+1})$ is a matrix and $b(\mathbf{u}^{n+1})$ is a vector field in the same Euclidean space. For the sake of computational efficiency and for avoiding the amplification of the singularities, Lemieux et al. used a first order approximation of the Gateaux derivative for the Jacobian of $F(\mathbf{u})$, applied to an arbitrary vector \mathbf{v} . To achieve a more accurate approximation for the Jacobian without sacrificing too much computational efficiency, here, we take advantage of the particular form of the nonlinear functional F and formulate, in (3.28), the derivatives of the linear part, namely, the action of the matrix A on the vector \mathbf{u}^{n+1} , and the quadratic nonlinearities contained in the vector \mathbf{b} , in closed form and use a second order approximation of the Gateaux derivative of $A(\mathbf{u})$, which contains the main singularity.

Numerical tests based on a synthetic exact solution for an augmented system of equations were conducted in Section 4. These tests allowed us not only to validate and verify that our computer program, written in FORTRAN 90 for better portability and compatibility with existing climate models, is functioning as desired but it also demonstrated that the solution is reproduced accurately. Table 4.2 shows that at a moderate resolution of 20 km in space and 10 min in time, the numerical error is well behaved and does not amplify too much, which perhaps pertains to the stability properties of the Crank–Nicolson method. Also as shown in Tables 4.3 and 4.4, the numerical solution seems to converge overall with the expected second order rate; the times at which the error does not decrease as much when passing from one grid resolution to the next, the error is most likely compensated with a rapid decrease when passing to the next grid. For instance, the mediocre convergence rate seen at day 5, in the L_2 norm, when going from the

moderate to the fine grid (or the negative rate associated with the L_∞ norm for that matter) is largely compensated by the effective rate of convergence, of about 4, seen during the passage from the coarse to the moderate grid. Since analytically speaking, the second order convergence is expected only in the limit of zero mesh size, this kind of behavior is not surprising, especially for a highly nonlinear system of equations like the SIME. More importantly, for the standard grid resolution of 20 km, as shown in Fig. 4.4, the nonlinear solver reached convergence within a maximum of 11 iterations per time step (the average of 8.4), which is a highly desirable feature in order to achieve an efficiently competitive sea ice model. However, the nonlinear solver fails to converge in a reasonable number of iterations at a few occasions, especially at the coarse resolution. Regardless, the conditional termination strategy introduced in Section 4.4, shows that formal convergence of the nonlinear solver is not always necessary, and the termination of the nonlinear solver whenever the residual begins to increase (by a certain amount) results in an equally accurate solution overall with significantly less nonlinear iteration counts (Figs. 4.9 and 4.10). It is worth stating that the sporadic behavior seen in Fig. 4.6b is likely associated with an “overshooting” of the Newton iterates, leading to the sudden increases in residual. In contrast to the conditional termination strategy, another method explored by the authors is the use of a damped Newton algorithm [31], limiting the size of each step taken by the nonlinear solver; a similar method was recently implemented by Lemieux et al. [32]. Numerical tests (shown and discussed in [24]) showed that smooth convergence can be achieved with this approach and that some combination of both Newton damping and conditional termination may be suitable to balance accuracy and computational efficiency.

In Section 4.3, the improved Jacobian approximation is tested against the first order Gateaux derivative approximation for the singular terms. As shown in Fig. 4.7b the performance of the JFNK solver was badly deteriorated both in terms of the number of nonlinear iterations and in terms of the deviation of the numerical solution from the target solution, especially after the first hour of simulation. With the first order approximation, the number of nonlinear iterations ranged between 60 and 160 during the first hour and the solver failed to converge within the imposed 200 iteration maximum during the second hour or so. The poor performance of the first order method is further illustrated by the plots in Fig. 4.8b. This stark difference in performance is in sharp contrast to what is reported in [22], where it is stated that when a centered difference approximation is applied to the entire Gateaux derivative, the added accuracy is negligible. The reason for this discrepancy is currently unknown, but considering that the second order approximation proved crucial for the convergence of our solver, careful investigation is required. Furthermore, as shown in Section 4.5.2 and Fig. 4.13, the second order method is proved to be robust for a large range of ϵ values ($10^{-8} \leq \epsilon \leq 10^{-4}$), while numerical tests (not shown) on the first order approach point to a significant sensitivity, agreeing with later work by Lemieux et al. [32,33], where a distinct sensitivity to ϵ for their first order approach is noted. The method reported here can be thought of as a *partially* Jacobian free (PJF) method, as we formulate the “simple” parts of the Jacobian and approximate the remaining terms. Additional work should be conducted to assess the effects of the PJF method; this may provide insight into why the second order approach proved so important in our tests. Additionally, to the best of the authors’ knowledge, all implementations by others of the JFNK solver for the SIME [19,22,32,33] incorporate some sort of preconditioning operation for the linear solver while the tests shown here and [24] do not. To assess the effect of this, a detailed study of the performance of the linear solver is required, which would also provide insight into the differences seen between methods for the Jacobian approximation.

The sensitivity to the coefficient γ_{nl} , used to fix the magnitude of the tolerance error for the convergence of the nonlinear solver, is tested in Section 4.5.1. As shown in Fig. 4.12, while the accuracy of the solution is fairly insensitive to this parameter the number of nonlinear iterations increased somewhat substantially, from around 8 iterations on average for the standard value of $\gamma_{nl} = 10$ to nearly 15 iteration with the tighter tolerance of $\gamma_{nl} = 0.01$. This suggests that the tightening of the nonlinear solver tolerance results only in unnecessary extra-computational time and that $\gamma_{nl} = 10$ seems to be optimal for the present test case at a 20 km grid resolution since the solver failed to converge with $\gamma_{nl} = 100$. However, the results in Fig. 4.12 suggest that $\gamma_{nl} = 100$ maybe the optimal tolerance coefficient at the coarse resolution. This particularly implies that the tolerance formula suggested in (3.32) is far from being universal and further research is needed in order to come up with a resolution aware error tolerance for the nonlinear solver.

In terms of mathematical analysis, very little is known about the sea ice momentum equations in (2.1). To the best of the authors’ knowledge, very few studies in this regard were reported in the literature and they are typically concerned with the stability of the SIME equations in the one dimensional case. Gray and Killworth [14] showed that the sea ice rheology equations with an elliptic yield curve, as utilized here, are linearly unstable due essentially to the lack of resistance of the ice to tensile stress. However, they argued that the equations remain nonlinearly stable because the divergent ice floes induce the ice strength to vanish and thus stabilizing the system. However, they warned the use of these equations as is in numerical simulations as the linear instabilities may excite grid scale waves that can pollute the numerical solution. Using an energy method, Dukowicz [34] showed that the equations are overall dissipative and thus stable, which is consistent with Gray and Killworth’s suggested nonlinear stability but it was not recognized by Dukowicz. More recently, Guba et al. [35] studied the well posedness of the SIME equation both in their original viscous-plastic version and in the case of a modified version using a pressure-replacement with a tensile cutoff. They found that, in the first case the SIME is well-posed when the flow divergence is below some threshold and ill-posed otherwise, while in the case with a tensile cutoff, the model is ill-posed whenever there is flow divergence. They also argue that well posedness does not guarantee stability and that statements such as “a bound on the energy in sea-ice models provides control over ill-posedness”, are flawed. Gray and Killworth [14] argued that a numerical viscosity is necessary in order to stabilize the discretized SIME equations and alternatively suggested a modification of the yield curve to allow a stress tensor, σ , which goes continuously to zero during a convergence stress by modifying the dependence of σ on the strain rate.

It is possible that the lack of quadratic convergence of the Newton solver and other difficulties encountered here are associated with this linear instability problem, reported in [14], and careful investigation is warranted in order to clarify this issue. While the Crank–Nicolson method is in general a more accurate representation of the governing equations, it is known to exhibit a marginal A-stability region [36] which is unable to damp the unstable modes inherent to the VP model, discussed above. This may explain why the backward Euler approach, which is known to be L-stable (its A-stability region extends well into the right portion of the complex plane), seems to perform better, in the sense that the convergence rate of the Newton’s method is closer to the quadratic limit [19,22]. In a practical sense, this difference in A-stability regions implies that the Crank–Nicolson approach is incapable of damping growing modes while the backward Euler method can. At coarser resolutions the added damping of the backward Euler approach should suffice to control the instability,

however in Gray and Killworth's work [14], it is also stated that the linear instability can become excited at finer grid scales and thus as the resolution is increased it is expected that the backward Euler method would also become incapable of damping the growing oscillations. A detailed comparison between the two methods is required to assess this, particularly a study utilizing the SIME. It is the authors' opinion that instead of relying on a less accurate, more diffusive scheme to control this inherent small scale instability of the VP equations, additional work should be conducted to create a fully second order scheme with better damping properties (for instance a backward differentiation formula (BDF) method [36]) or as suggested by Gray and Killworth [14], utilizing a modified yield curve that does not exhibit this linear instability. It would also be beneficial to gain additional insight into the noted instability by extending the stability analysis to two dimensions, as cross terms may play a role of either amplifying or damping this instability all together. For a working solution to this problem, as is done in [24], hyper-viscosity can also be added to the model and coupled with the Crank–Nicolson scheme in order to control the oscillations yet maintain the added accuracy of a fully second order scheme.

Acknowledgments

This work is part of C.S.'s master's thesis. The research of B.K. is partially supported by a discovery grant from the Natural Sciences and Engineering Council of Canada (RGPIN 2015-04288). C.S. is partially funded through this grant.

References

- [1] F. Scott, D. Feltham, A model of the three-dimensional evolution of arctic melt ponds on first-year and multiyear sea ice, *J. Geophys. Res.* 115 (2010).
- [2] D.K. Perovich, J.A Richter-Menge, Regional variability in sea ice melt in a changing Arctic, *Philos. Trans. Math. Phys. Eng. Sci.* 373 (2045) (2014) 20140165.
- [3] S. Juricke, T. Jung, Influence of stochastic sea ice parametrization on climate and the role of the atmosphere-sea ice-ocean interaction, *Phil. Trans. R. Soc. A* 372 (2014).
- [4] J.M.N.T. Gray, L.W. Morland, A two-dimensional model for the dynamics of sea ice, *Phil. Trans. R. Soc. Lond. A* 347 (1994) 219–290.
- [5] D.L. Feltham, Sea ice rheology, *Annu. Rev. Fluid Mech.* 40 (2008) 90–112.
- [6] W. Hibler, A dynamic thermodynamic sea ice model, *J. Phys. Oceanogr.* 9 (1979) 815–846.
- [7] S. Häkkinen, A constitutive law for sea ice and some applications, *Math. Model.* 9 (1987) 81–90.
- [8] M. Coon, G. Maykut, R. Pritchard, D. Rothrock, A. Thorndike, Modeling the pack ice as an elastic-plastic material, *AIDJEX Bull.* 24 (1974).
- [9] W. Hibler, A viscous sea ice law as a stochastic average of plasticity, *J. Geophys. Res.* 82 (1977) 3832–3938.
- [10] W.J. Campbell, The wind-driven circulation of ice and water in a polar ocean, *J. Geophys. Res.* 70 (14) (1965) 3279–3301.
- [11] W.J. Campbell, A. Rasmussen, Latest experiments with ice rheology, *J. Geophys. Res.* 70 (14) (1965) 3279–3301.
- [12] J.W. Glen, Thoughts on a viscous model for sea-ice, *AIDJEX Bull.* 2 (1970) 18–27.
- [13] R.S. Pritchard, M.D. Coon, M.G. McPhee, Simulation of sea ice dynamics during AIDJEX, *AIDJEX Bull.* 34 (1970) 73–93.
- [14] J.M.N.T. Gray, P.D. Killworth, Stability of the viscous-plastic sea ice rheology, *J. Phys. Oceanogr.* 25 (1995) 971–978.
- [15] J. Zhang, W. Hibler, On an efficient numerical method for modeling sea ice dynamics, *J. Geophys. Res.* 102 (1997).
- [16] J.-F. Lemieux, B. Tremblay, Numerical convergence of viscous-plastic sea ice models, *J. Geophys. Res.* 114 (2009).
- [17] E. Hunke, J. Dukowicz, An elastic-viscous-plastic model for sea ice dynamics, *J. Phys. Oceanogr.* 27 (1997).
- [18] L. Girard, S.B. Bouillon, J. Weiss, D. Amitrano, T. Fichefet, V. Legat, A new modelling framework for sea-ice mechanics based on elasto-brittle rheology, *Ann. Glaciol.* 52 (2011) 123–132.
- [19] J.-F. Lemieux, D. Knoll, B. Tremblay, D. Holland, M. Losch, A comparison of the Jacobian-free Newton-Krylov method and the EVP model for solving the sea ice momentum equation with a viscous-plastic formulation: A serial algorithm study, *J. Comput. Phys.* 231 (2012) 5926–5944.
- [20] E. Hunke, Y. Zhang, A comparison of sea ice dynamics models at high resolutions, *Mon. Weather Rev.* 127 (1998).
- [21] E.C. Hunke, Viscous-plastic sea ice dynamics with the EVP model: Linearization Issues, *J. Comput. Phys.* 170 (2001).
- [22] J.-F. Lemieux, B. Tremblay, J. Sedlacek, P. Tupper, S. Thomas, D. Huard, J.-P. Auclair, Improving the numerical convergence of viscous-plastic sea ice models with the Jacobian-free Newton-Krylov method, *J. Comput. Phys.* 229 (2010) 2840–2852.
- [23] L.-B. Tremblay, L. Mysak, Modeling sea ice as a granular material, including the dilatancy effect, *J. Phys. Oceanogr.* 27 (1997).
- [24] C. Seinen, A fast and efficient solver for viscous-plastic sea ice dynamics, *The University of Victoria*, 2017 (master's thesis).
- [25] M. Losch, S. Danilov, On solving the momentum equations of dynamic sea ice models with implicit solvers and the elastic-viscous-plastic technique, *Ocean Modell.* 41 (2012).
- [26] A. Bourlioux, B. Khouider, A rigourous asymptotic perspective on the large scale simulations of turbulent premixed flames Multiscale Modeling and Simulation, *Multiscale Model. Simul.* 6 (2007) 287–307.
- [27] D. Knoll, D. Keyes, Jacobian-free Newton-Krylov methods: A survey of approaches and applications, *J. Comput. Phys.* 193 (2004).
- [28] Y. Saad, *Iterative Methods for Sparse Linear Systems*, second ed., SIAM, 2003.
- [29] M.D.L. Chevrotiere, *Stochastic and Numerical Models for Tropical Convection and Hadley-Monsoon Dynamics*, The University of Victoria, 2015 (Ph.D. thesis).
- [30] P. Roache, Quantification of uncertainty in computational fluid dynamics, *Annu. Rev. Fluid Mech.* 29 (1997).
- [31] L.P. Saumier, M. Agueh, B. Khouider, An efficient numerical algorithm for the L^2 optimal transport problem with periodic densities, *IMA J. Appl. Math.* 80 (2015) 135–157.
- [32] J.-F. Lemieux, D. Knoll, M. Losch, C. Girard, A second-order accurate in time Implicit-Explicit (imex) integration scheme for sea ice dynamics, *J. Comput. Phys.* (2014).
- [33] M. Losch, A. Fuchs, J. Lemieux, A. Vanselow, A parallel Jacobian-free Newton-Krylov solver for a coupled sea ice-ocean model, *J. Comput. Phys.* 257 (2014).
- [34] J.K. Dukowicz, Comments on “Stability of the viscous-plastic sea ice rheology”, *J. Phys. Oceanogr.* 27 (1997) 480–481.
- [35] O. Guba, J. Lorenz, D. Sulsky, On well-posedness of the viscous-plastic sea ice model, *J. Phys. Oceanogr.* 43 (2013) 2185–2199.
- [36] R. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Dime-Dependent Problems*, SIAM, 2007.